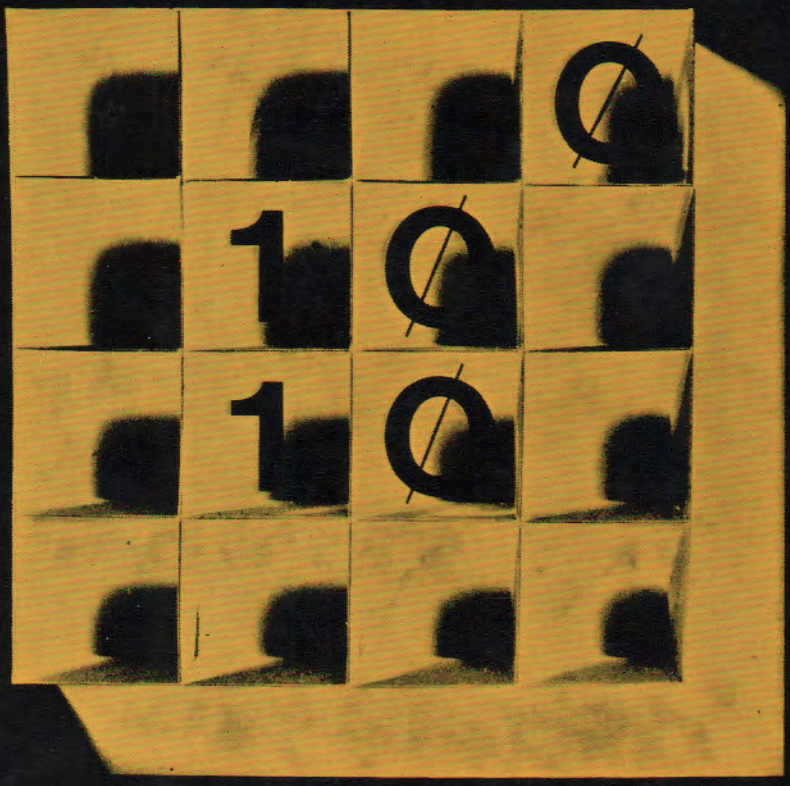
A. PETITCLERC

TRATADO DE ORDENADORES

TOMO I

álgebra lógica, aritmética binaria, algoritmos



editores técnicos asociados, s. a.
BARCELONA

GARANTÍA
EL LIBRO TECNICO
TOMÁS MORALES, 44
36 27 91 - 36 77 06

Tratado de ordenadores

TOMO 1

Algebra lógica
Aritmética binaria
Algoritmos

A. PETITCLERC
Ingeniero de la Compañía IBM Francia
Miembro de la "Société Française des Électroniciens et des Radioélectriciens"

Tratado de ordenadores

TOMO 1

Algebra lógica
Aritmética binaria
Algoritmos

editores técnicos asociados, s. a.
Maignón, 26 – Barcelona - 12 – España
1 9 7 4

Primera edición española, traducida del francés por
EDUARDO BUZZANCA CAÑELLAS
Dr. Ingeniero Industrial

Título de la obra original:

TRAITE DES ORDINATEURS

Publicada por

DUNOD – París

© EDITORES TECNICOS ASOCIADOS, S.A. – Barcelona 1974

Depósito legal B. 22.236-1974
ISBN (obra completa) 84-7146-141-2
ISBN (tomo I) 84-7146-142-0

Impreso en España
Printed in Spain

ROMARGRAF, S.A. – Sants, 387 – Barcelona - España

Indice de materias

Introducción IX

Capítulo 1. ***Algebra de Boole*** 1

Introducción 3
Operaciones fundamentales en álgebra de Boole 3
Tablas de verdad y teoremas fundamentales 8
El teorema de De Morgan: inversión de una expresión "booleana" 16
Algunos circuitos lógicos importantes: O exclusivo, NO-O y NO-Y 19
Funciones realizadas por dos variables "booleanas" 41
Aplicación de la tabla de verdad: circuitos de codificación y decodificación 48
Simplificación de los circuitos 52
Los circuitos lógicos de memoria 56
Básculas J-K 58
Básculas R-S 59
Básculas R-S-T 60

Capítulo 2. ***Los diagramas de Karnaugh*** 63

Introducción 65
Definición del diagrama de Karnaugh. Introducción de una expresión "booleana" en un diagrama de Karnaugh 65
Lectura del diagrama de Karnaugh. Simplificación de las funciones 75
Condiciones prohibidas 100
Circuitos de múltiples salidas 109
Conclusión sobre el método de los diagramas de Karnaugh 119
El método de Quine y McCluskey 119

Capítulo 3. ***Aritmética binaria*** 131

Generalidades 133
Consideraciones sobre el sistema decimal 133
El sistema binario 135

Conversión decimal binario 139
Naturaleza de los números tratados por los registros 141
Representación binaria de los números con signo 142
Adición y substracción en el caso en que los números se presenten con módulo y signo 146
Suma y resta en el caso en que se representan los números negativos por su complemento a 1 153
Adición y substracción en el caso en que los números negativos se representan por su complemento a 2 158
Tamaño de los números en los registros 160
Desplazamiento de los números en los registros 163
La multiplicación en las calculadoras. Generalidades 165
El problema de los signos 168
Desarrollo del algoritmo de la multiplicación de dos números expresados en valor absoluto y en signo 169
Multiplicación por el método de Booth 172
La división automática 182
Las calculadoras decimales 207
Multiplicación decimal 221
La división decimal 225
Cálculos en coma flotante 227
Diferentes códigos utilizados para representar un número decimal 229
Código biquinario 238

Bibliografía 241

Introducción

Esta obra se propone describir la teoría y los principios generales de funcionamiento de los ordenadores. Se dirige a los ingenieros y técnicos de diversas disciplinas que deseen adquirir un profundo conocimiento sobre la materia, tanto en el dominio de los circuitos del ordenador como en su concepción. El lector de esta obra no necesita un conocimiento especial sobre informática. Se ha redactado con un deseo; ser lo suficientemente simple para poder utilizarse sin profesor. Su carácter didáctico permitirá que los lectores no técnicos que deseen más claridad en este dominio, lo aborden sin dificultades.

El principiante que inicia por primera vez el estudio de un ordenador sólo dispone de obras de divulgación o muy especializadas, que tratan de forma superficial o demasiado teórica un aspecto muy particular del funcionamiento de un ordenador y de su utilización. Resulta muy difícil realizar la síntesis entre los diferentes elementos del ordenador. Nos proponemos cubrir esta laguna dando a los lectores un texto que trate de forma detallada todas las principales funciones de un ordenador para iniciar a continuación el estudio de su organización interna y su utilización mediante un programa. Los conocimientos de electrónica se reducen al mínimo, el estudio de los circuitos se efectúa de forma lógica sin entrar en el detalle técnico de su realización. Esto es objeto de un curso de semiconductores. El lector que esté interesado por este dominio podrá consultar los excelentes tratados que sobre este tema existen en el mercado.

El ordenador tiene un lugar cada vez más importante en nuestra sociedad. Guía *los aviones* y satélites, controla los viajes interplanetarios, el tráfico de nuestras carreteras, la producción de nuestras fábricas, asegura la gestión en nuestras empresas. El día de mañana enseñará a nuestros hijos las materias más difíciles. Esta invasión del ordenador en nuestra vida cotidiana modificará las actividades de muchos de nosotros. Algunos construirán estos ordenadores, otros trabajarán estableciendo los programas específicos a cada problema que el ordenador deba resolver.

La presente obra, dividida en dos tomos, es el contenido de un curso que el autor da a los ingenieros y técnicos de la compañía I.B.M. en Francia para el perfeccionamiento científico de su personal.

En el tomo 1 se hallará la exposición de los métodos de análisis de los circuitos de conmutación y, en el tomo 2, el estudio de los circuitos de un ordenador y de su organización interna. El primer capítulo del tomo 1 se dedica al estudio del álgebra de Boole, la herramienta por excelencia de los informáticos. Numerosos ejemplos ilustran el método. El segundo capítulo se dedica al estudio de los diagramas de Karnaugh, que representan un medio poderoso y elegante de simplificación de los circuitos lógicos más complejos. Se estudian otros métodos tales como los de Quine y Mc. Cluskey. El tercer capítulo trata la aritmética binaria. El lector hallará en este capítulo el desarrollo de los algoritmos de las distintas operaciones realizadas por el ordenador.

El tomo 2 trata las principales funciones del ordenador y sus circuitos. El capítulo primero se dedica al estudio de los codificadores, contadores y registros, así como los circuitos que transfieren la información de registro a registro en serie o en paralelo. El capítulo segundo estudia con mucho detalle los diferentes tipos de adicionadores y substractores así como las importantes funciones del acumulador. El tercer capítulo estudia la memoria del ordenador. Se dedica una atención especial a la determinación de los circuitos de selección y direccionamiento que generalmente es un punto difícil para muchos de los principiantes. El siguiente capítulo se dedica a la unidad aritmética y lógica, particularmente a los circuitos que realizan los ciclos de instrucción y ejecución de las principales operaciones aritméticas. El tomo 2 termina con un capítulo dedicado a la organización interna del ordenador. El lector puede seguir de esta manera, a partir de un código previamente definido, la codificación de las instrucciones y su ejecución en los diferentes registros y circuitos del ordenador. Terminado el tomo 2, el lector estará en condiciones de conocer el funcionamiento de un ordenador y la forma de determinar sus circuitos.

El autor da las gracias a la compañía I.B.M. Francia, por la favorable acogida dispensada a la obra, así como a todos sus alumnos por el interés testimoniado continuamente durante la enseñanza de este curso.

A. PETITCLERC

Capítulo 1

Algebra de Boole

1. Introducción

En 1854 el físico inglés George Boole publicó su obra *"Theory of Thought"*. De acuerdo con su teoría, los procesos mentales en el hombre resultan de la asociación de elementos simples que responden a dos alternativas: sí - no. Para exponer sus ideas desarrolla un método de análisis, llamado posteriormente álgebra de Boole o álgebra lógica. En este álgebra las variables sólo pueden tomar dos valores 0 y 1, sin pasar por valores intermedios.

Este método de análisis considerado durante mucho tiempo como una curiosidad, ha adquirido difusión considerable después de la aparición de las calculadoras numéricas.

En efecto, ellas se construyen con la ayuda de elementos binarios que sólo pueden tomar dos estados eléctricos, a los que se les atribuye los valores 0 y 1.

En las primeras máquinas, estos elementos se representaban mediante relés mecánicos y un relé sólo puede estar abierto o cerrado. Más tarde las válvulas electrónicas y después los transistores han sustituido a los relés pero el modo de funcionamiento es el mismo: una válvula o un transistor está bloqueado o saturado, lo que equivale a los dos estados: abierto o cerrado de los relés mecánicos.

Actualmente puede decirse, sin exageración, que el álgebra de Boole es la herramienta, por excelencia, de los lógicos que estudian las calculadoras numéricas. Permite establecer los circuitos lógicos más simples necesarios para las transferencias de datos, el control y la ejecución de operaciones aritméticas.

El objeto de esta obra es hacer conocer a los lectores las bases de este cálculo y su utilización en la realización de los circuitos de las calculadoras.

2. Operaciones fundamentales en álgebra de Boole

Una variable en álgebra de Boole, si varía, sólo puede tomar los valores 0 ó 1. Si es 0 pasará a 1 y si es 1 pasará a 0. Hay tres operaciones fundamentales en el álgebra de Boole.

- la inversión u operación "NO".
- la multiplicación lógica u operación "Y".
- la adición lógica u operación "O".

Estas operaciones se designan en la literatura anglosajona por "NOT", "AND" y "OR". Estas operaciones se realizan mediante tres circuitos elementales, llamados "NO", "Y" y "O". Todos los circuitos de las calculadoras se realizan a partir de estos circuitos elementales o de sus combinaciones. Como combinaciones parti-

cularmente importantes, hay que señalar los siguientes circuitos:

- circuito "O exclusivo".
- circuito "NO-O" o circuito "NOR".
- circuito "NO-Y" o circuito "NAND".

La división no existe en el álgebra de Boole, una división por 0 no tendría ninguna significación.

2.1 La inversión

Como sólo existen dos valores 0 y 1 en el álgebra de Boole, el inverso de una variable es el otro valor de la variable.

Si el valor de la variable es 1, su inverso es 0 y si es 0 su inverso es 1. Esta operación se representa colocando una barra por encima de la variable a invertir. Así, de acuerdo con la definición, la inversa de A será $\overline{A}$ con las dos posibilidades:

$$\boxed{\begin{array}{l}\overline{1} = 0 \\ \overline{0} = 1\end{array}} \quad . \tag{1}$$

El circuito *lógico* que realizará esa operación se representa, fig. 1, mediante un cuadrado que contiene la letra I.

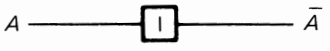


Fig. 1. *Inversión de una variable.*

La figura 2 da la representación de la operación inversión bajo la forma de interruptores.

Fig. 2. *Realización de la operación de inversión mediante un interruptor.*

2.2 Multiplicación lógica u operación Y

Esta operación se efectúa sobre dos o más variables. Se define por:

$$\boxed{\begin{array}{l} 0.0 = 0 \\ 0.1 = 0 \\ 1.0 = 0 \\ 1.1 = 1 \end{array}}\ , \qquad (2)$$

lo que equivale a la siguiente definición: la multiplicación lógica de dos variables "booleanas" es otra variable "booleana" que toma el valor 1 si, y solamente si, las dos variables son 1. En los restantes casos es 0. La multiplicación lógica de dos variables "booleanas" da lugar a $2^2 = 4$ combinaciones (2). Esta operación se llama multiplicación lógica por analogía con la multiplicación clásica.

En la figura 3 se da la representación del circuito lógico que realiza esta operación. El producto lógico de dos variables A y B se representa por AB.

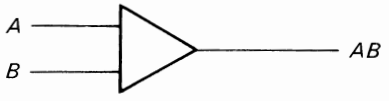


Fig. 3. *Representación de un circuito que realiza la multiplicación lógica de dos variables A y B.*

La salida de este circuito será 1 si la primera y la segunda variable son simultáneamente 1, de aquí el nombre de circuito Y, que se le da a este circuito. Podemos generalizar este circuito y considerar una multiplicación de n términos. Una representación equivalente consiste en asociar n interruptores *en serie*: Sólo habrá paso de corriente (1) si, y solamente si, el primer interruptor, Y el segundo, Y el tercero... están cerrados (1). Es suficiente que uno de ellos esté abierto (0) para que la salida sea cero (ausencia de corriente), figura 4.

La multiplicación lógica es conmutativa:

$$AB = BA\ , \qquad (3)$$

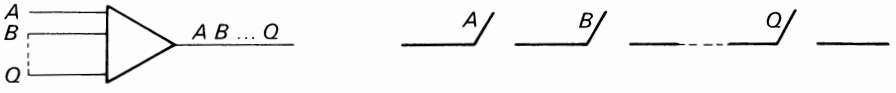


Fig. 4. *Circuito Y con varias variables y representación bajo la forma de interruptores en serie.*

también es asociativa, es decir:

$$A(BC) = (AB)\,C = (AC)\,B\,. \tag{4}$$

2.3 Suma lógica

La suma lógica de dos variables binarias tiene $2^2 = 4$ combinaciones:

$$\begin{array}{l} 0 + 0 = 0 \\ 0 + 1 = 1 \\ 1 + 0 = 1 \\ 1 + 1 = 1 \end{array}$$

Esta operación se llama suma lógica por analogía con la suma clásica: sólo la última suma es diferente. El resultado de la suma lógica es 1, si es uno una de las variables, de donde procede el nombre de circuito O que se da al circuito. La representación del circuito lógico que realiza la función O es la de la figura 5.

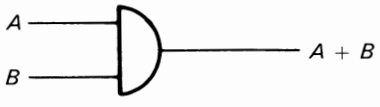


Fig. 5. *Representación de un circuito que realiza la suma lógica de dos variables A y B.*

De una forma más general si A y B son dos variables booleanas la suma lógica se representa por $A + B$.

El circuito de la figura 5 puede tener n entradas. En este caso la salida tendrá el valor 1, si cualquiera o la totalidad de las entradas tiene valor 1. Será 0 si la totalidad de las entradas es cero. La suma lógica puede representarse (Fig. 6) por la asociación de n interruptores en paralelo. Tendremos corriente en la salida (1) si uno cualquiera o todos los interruptores están cerrados (1). La salida será 0, si todos los interruptores están abiertos 0.

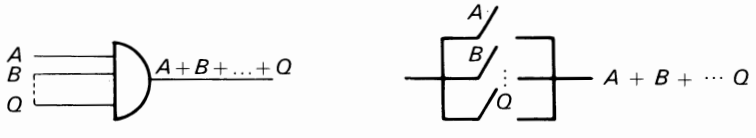


Fig. 6. *Circuito O con varias variables y representación en forma de interruptores en paralelo.*

La suma lógica es conmutativa:

$$A + B + C = A + C + B = B + A + C . \tag{5}$$

Es asociativa:

$$A + B + C = (A + B) + C = A + (B + C) . \tag{7}$$

Nota. Antes de efectuar el producto lógico de varias variables, una de las cuales es un inversor, se efectuará previamente la inversión de esta variable.

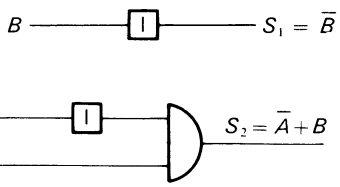
Ejemplo 1. *¿Cuál es el valor de*

$$X = 1 . 1 . \overline{1} . 1 . 1 \ ?$$

Se invierte primeramente el término barrado.

$$X = 1 . 1 . 0 . 1 . 1 = 0 .$$

Ejemplo 2. *¿Cuáles han de ser los valores A y B, para que las salidas de los dos circuitos presentados a continuación tengan el mismo valor 1?*



Obtendremos para el primer circuito:

$$S_1 = \overline{B}$$

y para el segundo:

$$S_2 = \overline{A} + B .$$

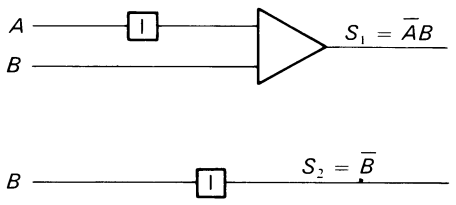
Para que S_1 sea 1 es necesario que $\overline{B} = 1$, es decir $B = 0$.
Llevemos este valor a S_2 y obtendremos

$$S_2 = \overline{A} = 1 ,$$

es decir $A = 0$. Luego ha de cumplirse la condición:

$$A = B = 0$$

Ejemplo 3. *¿Cuáles son los valores de A y B, para que las salidas de los dos circuitos que van a continuación sean iguales?*



Obtendremos sucesivamente:

$$S_1 = \bar{A}B ,$$
$$S_2 = \bar{B} .$$

Las dos salidas son iguales si

$$\bar{A}B = \bar{B} .$$

Atribuyendo sucesivamente a A y B los valores 0 y 1. La relación anterior sólo se satisfará si:

$$A = B = 1 .$$

En este momento $\bar{A}B = \bar{B} = 0$.

3. Tablas de verdad y teoremas fundamentales

Vamos a demostrar en este punto, los distintos teoremas fundamentales del álgebra de Boole, mediante una tabla que expresa los valores de la función estudiada en función de las diferentes combinaciones de estas variables. Esta tabla se llama tabla *de verdad.* Se divide en tres partes. En la parte situada a la izquierda se dan todas las posibles combinaciones de las variables. Así, una expresión con 2, 3, ...,n variables da lugar a 2^2, 2^3, ..., 2^n combinaciones. En la parte central se expresa la operación estudiada en función de las variables. En la parte derecha se trata de obtener una expresión equivalente más simplificada.

Teorema 1

$$A + 0 = A .$$

Se desea obtener la suma lógica de una variable A y de una variable B que conserva su valor 0. En la parte de la izquierda de la tabla se llevan los distintos valores de A, 0 y 1. En la parte central se hace la suma lógica $A + 0$ y en la parte de la derecha, se coloca el resultado. Comparando todas las líneas de la columna A con todas las líneas de la columna $A + 0$ se observará que $A + 0$ es equivalente a A.

A	$A + 0$	A
0	0	0
1	1	1

Podemos escribir:

$$A + 0 = A \qquad (8)$$

que se puede representar mediante el esquema eléctrico equivalente de la figura 7:

Fig. 7. *Equivalente eléctrico de la relación A + 0.*

Como se trata de una suma lógica tendremos el equivalente de la asociación en paralelo de un interruptor A y de un circuito abierto.

Se comprueba que se obtiene en la salida el valor "booleano" atribuido a A según que el interruptor esté abierto o cerrado.

Teorema 2

$$A + 1 = 1 .$$

Esta relación puede obtenerse directamente de la definición de la suma lógica de dos términos. En efecto, es suficiente que un término sea igual a 1 para que la suma sea igual a 1. La tabla de verdad se representa de la siguiente forma:

A	$A + 1$	1
0	1	1
1	1	1

Obtendremos la relación:

$$A + 1 = 1 . \qquad (9)$$

El esquema eléctrico equivalente es el de la figura 8:

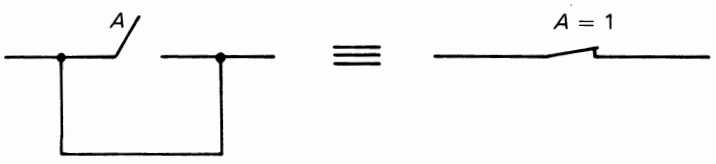


Fig. 8. *Equivalente eléctrico de relación A + 1 = 1.*

Teorema 3

$$A + A = A .$$

Esta relación es muy importante pues permite simplificar considerablemente las expresiones "booleanas". Obtenemos la siguiente tabla de verdad.

A	$A + A$	A
0	0	0
1	1	1

de donde se saca, mediante examen, la relación:

$$A + A = A \,. \tag{10}$$

El esquema eléctrico equivalente es el de la figura 9.

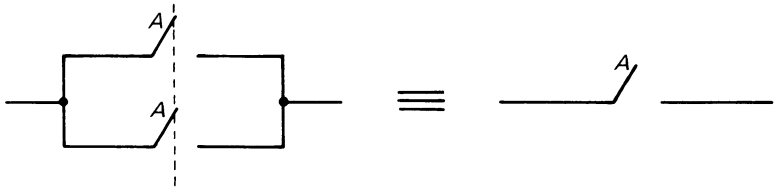


Fig. 9. *Equivalente eléctrico de la relación* $A + A = A$.

Teorema 4

$$A + \overline{A} = 1 \,.$$

$\overline{A}$ representa la inversa o complemento de A, es decir todo lo que no es A. La suma de A y de todo lo que no es A, es obligatoriamente 1. A continuación se da la tabla de verdad así como el equivalente eléctrico (Fig. 10).

A	$A + \overline{A}$	1
1	1	1
0	1	1

Del examen de la tabla se obtiene:

$$A + \overline{A} = 1 \tag{11}$$

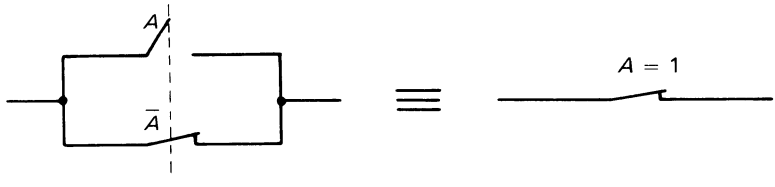


Fig. 10. *Equivalente eléctrico de la relación* $A + \overline{A} = 1$.

Se observa (Fig. 10) que, cualquiera que sea el valor atribuido a A, habrá siempre un interruptor cerrado. La salida será siempre 1.

Resumamos estos resultados:

$$\boxed{\begin{aligned} A + 0 &= A \\ A + 1 &= 1 \\ A + A &= A \\ A + \overline{A} &= 1 \end{aligned}} \quad . \tag{12}$$

Estas relaciones generalizan la operación de suma lógica.

Teorema 5

$$A.0 = 0 .$$

El producto lógico de una variable "boolena" por el valor 0 es nulo. Esto es una consecuencia directa de la definición de la multiplicación lógica. El examen de la tabla de verdad que se indica a continuación:

A	$A.0$	0
1	0	0
0	0	0

nos permite escribir la relación:

$$A.0 = 0 \tag{13}$$

lo que corresponde al esquema eléctrico equivalente de la figura 11*a*, donde se tiene colocado en serie el interruptor A y un circuito abierto. Cualquiera que sea la posición del interruptor la corriente, en la salida, será siempre nula. Tendrá constantemente el valor 0.

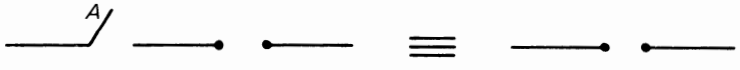


Fig. 11a. *Equivalente eléctrico de la relación A.0 = 0.*

Teorema 6

$$A.1 = A .$$

Esta relación corresponde a la asociación en serie de un interruptor A y de un cortocircuito (Fig. 11*b*). Lo que equivale a tener solamente un interruptor en el circuito.

La tabla de verdad es la siguiente:

A	$A.1$	A
0	0	0
1	1	1

El examen de esta tabla nos permite escribir

$$A.1 = A \tag{14}$$

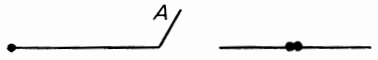


Fig. 11b. *Equivalente eléctrico de la relación A.1 = A.*

Teorema 7

$$A.A = A\ .$$

El esquema equivalente de la fig. 12 corresponde a la asociación en serie de dos interruptores controlados simultáneamente. Esto es equivalente a tener un solo interruptor en el circuito. La tabla de verdad verifica adecuadamente esta conclusión.

A	$A.A$	A
0	0	0
1	1	1

El examen de la tabla da:

$$A.A = A\ . \tag{15}$$

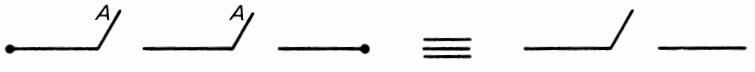


Fig. 12. *Equivalente eléctrico de la relación A.A = A.*

Teorema 8

$$A.\overline{A} = 0\ .$$

El producto de una variable "booleana" por su inversa es nulo.

El equivalente eléctrico se muestra en la figura 13. Cuando un interruptor está abierto, el otro está cerrado y recíprocamente, pero siempre uno de los dos inte-

rruptores está abierto: en salida no aparece ninguna corriente. Lo que equivale a 0.

La tabla de verdad es la siguiente:

A	$\bar{A}$	$A.\bar{A}$	0
0	1	0	0
1	0	0	0

El examen línea a línea de esta tabla nos da la relación:

$$A.\bar{A} = 0 \tag{16}$$

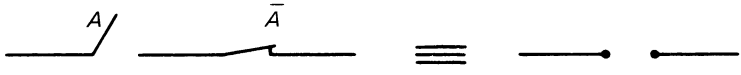


Fig. 13. *Equivalente eléctrico de la expresión $A.A = 0$.*

Resumamos estos resultados.

$$\begin{aligned} A.0 &= 0 \\ A.1 &= A \\ A.A &= A \\ A.\bar{A} &= 0 \end{aligned} \tag{17}$$

Estos resultados generalizan la operación de multiplicación lógica. Veamos ahora algunos casos más complicados, con dos variables.

Teorema 9

$$A + AB = A\,.$$

En las expresiones más complicadas se hace necesaria la utilización intensiva de *la tabla* de verdad. En el caso que nos interesa tenemos dos variables, con cuatro combinaciones; la tabla de verdad es la siguiente:

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Para establecer todas las combinaciones posibles de dos variables AB se atribuye al principio los valores 0 y 1 a la variable B a los que se les hace corresponder el valor 0 de la variable A. A continuación se repiten los valores 0 y 1 de la variable B a las que se hace corresponder ahora el valor 1 de la variable A.

Esto es lo que se ha hecho en las dos primeras columnas de la tabla de verdad. La tercera columna da el valor del término AB y la cuarta columna da la expresión $A + AB$. El examen de todas las líneas de la tabla nos muestra la identidad de las columnas A y $A + AB$. Obtenemos, la siguiente relación:

$$A + AB = A \,. \tag{18}$$

Esta relación es muy importante y se utiliza con frecuencia. El circuito eléctrico equivalente es el de la figura 14, que posee dos ramas en paralelo pues se trata de una adición lógica: una de las ramas contiene el interruptor A solo, la otra los interruptores A y B en serie.

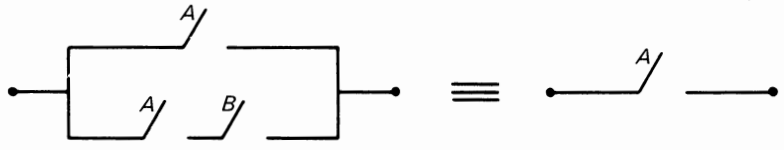


Fig. 14. *Equivalente eléctrico de la relación* $A + AB$.

Teorema 10

$$A + \bar{A}B = A + B \,.$$

En esta expresión tenemos igualmente dos variables A y B y cuatro combinaciones posibles. Desde el principio estableceremos la tabla de verdad.

Examinando esta tabla se obtendrá que la expresión $A + \bar{A}B$ es igual a $A + B$. Podemos escribir:

A	B	$\bar{A}B$	$A + \bar{A}B$	$A + B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

$$A + \bar{A}B = A + B \,. \tag{19}$$

Esta relación es también muy importante y se utiliza con mucha frecuencia. El esquema eléctrico equivalente es el de la figura 15:

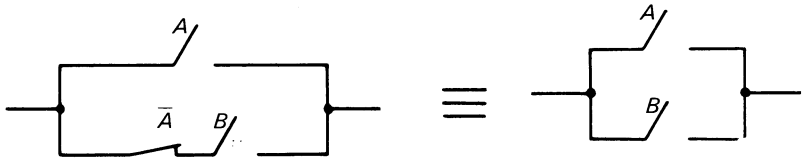


Fig. 15. *Equivalente eléctrico de la relación* $A + \bar{A}B$.

Teorema 11

$$A(B + C) = AB + AC .$$

Este teorema representa la propiedad distributiva del producto lógico. Es idéntica a la del álgebra ordinaria y permite el desarrollo de ciertas expresiones. Para verificar este teorema se utiliza la siguiente tabla de verdad de tres variables A, B, C:

A	B	C	$B + C$	$A(B + C)$	$AB + AC$
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	0	0
1	0	1	1	1	1
1	1	0	1	1	1
1	1	1	1	1	1

El examen completo de esta tabla pone de manifiesto la identidad

$$A(B + C) = AB + AC \tag{20}$$

utilizada constantemente tanto en el álgebra de Boole, como en el álgebra ordinaria. Utilizando los interruptores, el esquema eléctrico equivalente es el de la figura 16:

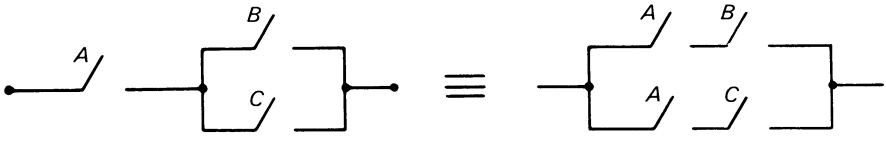


Fig. 16. *Equivalente eléctrico de la relación* $A\,(B + C) = AB + AC$.

Teorema 12

$$A(A + B) = A .$$

Este teorema se demuestra apoyándose en los teoremas 9 y 11:

$$A(A + B) = AA + AB .$$

A partir de (17) sabemos que $AA = A$, de donde:

$$A(A + B) = A + AB$$

y según el teorema 9:

$$A(A + B) = A . \tag{21}$$

Teorema 13

$$(A + B)(A + \overline{B}) = A .$$

Este teorema se demuestra apoyándose en los teoremas precedentes. Tenemos:

$$(A + B)(A + \overline{B}) = AA + A\overline{B} + AB + B\overline{B}$$

y según (16) y (18):

$$(A + B)(A + \overline{B}) = A . \tag{22}$$

Esta relación se utiliza frecuentemente para la simplificación de circuitos.

Teorema 14

La doble inversión $\overline{\overline{A}} = A$.

Este teorema se demuestra inmediatamente volviendo a la definición de inversión. Demos a A el valor de 1, obtendremos $\overline{A} = 0$ y una segunda inversión da el valor $\overline{\overline{A}} = 1$, es decir A. Si $A = 0, \overline{A} = 1$ y una segunda inversión da $\overline{\overline{A}} = 0$, es decir A. En todos los casos podemos escribir:

$$\overline{\overline{A}} = A . \tag{23}$$

4. El teorema de De Morgan: inversión de una expresión «booleana»

El teorema de De Morgan permite calcular expresiones de la forma

$$\overline{A\,B\,C\,D \ldots Q} \quad \text{o} \quad \overline{A + B + C + D + \cdots + Q} ,$$

los términos A, B, C, D, ... Q pueden agrupar varios términos. El teorema de De Morgan permite calcular la inversa de cualquier expresión "booleana". En el primer caso el producto invertido se transforma en suma de términos inversos. En el segundo caso la inversión de la suma se transforma en producto de términos inversos. Tomemos el primer caso limitándonos a dos términos y demostremos mediante la tabla de verdad que $\overline{AB} = \bar{A} + \bar{B}$. La tabla de verdad es la siguiente:

A	B	$\overline{AB}$	$\bar{A} + \bar{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Comparando las dos últimas columnas línea a línea, se ve claramente la identidad de las dos expresiones:

$$\overline{AB} = \bar{A} + \bar{B} . \tag{24}$$

Mediante iteraciones sucesivas puede extenderse el teorema a n términos:

$$\overline{A\,B\,C\,D \ldots Q} = \bar{A} + \bar{B} + \bar{C} + \bar{D} + \cdots + \bar{Q} . \tag{25}$$

Veamos ahora el segundo caso, limitémosle también a dos términos y demostremos que $\overline{A + B} = \bar{A}\,\bar{B}$.

A continuación tenemos la tabla de verdad:

A	B	$\overline{A + B}$	$\bar{A}\bar{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

La comparación de las dos últimas columnas línea a línea nos permite escribir:

$$\overline{A + B} = \bar{A} . \bar{B} . \tag{26}$$

Se puede generalizar a n términos mediante sucesivas iteraciones:

$$\overline{A + B + C + D + \cdots + Q} = \bar{A}\bar{B}\bar{C}\bar{D} \ldots \bar{Q} . \tag{27}$$

Se observará que las expresiones (25) y (27) pueden expresarse mediante un teorema único:

Teorema

Para obtener la inversa de una función booleana se efectúan las siguientes operaciones:

- *efectuar, si es necesario, las sustituciones para llevar la función a invertir a una de las formas precedentes,*
- *invertir todas las variables,*
- *cambiar las sumas por productos y los productos por sumas.*

Ejemplo 1. *Calcular la siguiente expresión mediante el teorema de De Morgan:*

$$X = \overline{AB + C} = \overline{AB}.\overline{C}.$$

En esta expresión se considera el producto AB como una sola variable, para llegar a una expresión de la forma (27). Aplicando nuevamente el teorema de De Morgan al producto AB obtenemos finalmente:

$$X = \overline{AB + C} = (\overline{A} + \overline{B}).\overline{C} = \overline{A}\overline{C} + \overline{B}\overline{C}.$$

Ejemplo 2. *Transformar la siguiente expresión:*

$$X = \overline{A + \overline{BC} + \overline{C}D}.$$

Utilizando el teorema de De Morgan obtenemos:

$$X = \overline{A}.\overline{\overline{BC}}.\overline{\overline{C}D},$$

es decir, como $\overline{\overline{BC}} = BC$:

$$X = \overline{A}.BC.\overline{\overline{C}D}.$$

Aplicando nuevamente el teorema de De Morgan al último término:

$$X = \overline{A}BC(C + \overline{D}) = \overline{A}BC + \overline{A}BC\overline{D},$$

teniendo en cuenta que $\overline{A}BCC = \overline{A}BC$; de donde según (18):

$$X = \overline{A}BC.$$

4.1 Caso de una inversión superpuesta a otra inversión.

Si en una expresión "booleana" superponemos una inversión a otra inversión, el resultado es independiente del orden de las inversiones.

Ejemplo. *Transformar la siguiente expresión:*

$$X = \overline{A + \overline{BC} + \overline{CD + B}},$$

mediante el teorema de De Morgan.

Podemos comenzar eliminando la barra superior o la barra inferior. Eliminaremos la barra superior:

$$X = \bar{A}.\overline{\bar{B}C}.\overline{\overline{CD + B}},$$

es decir:

$$X = \bar{A}.(B + \bar{C}).(CD + B) = \bar{A}B,$$

que es el resultado buscado.

Procedamos ahora eliminando la barra interior:

$$X = \overline{A + \bar{B}C + (\overline{CD}.\bar{B})},$$

es decir:

$$X = \overline{A + \bar{B}C + (\bar{C} + \bar{D})\bar{B}}.$$

Eliminemos la barra superior:

$$X = \bar{A}.\overline{\bar{B}C}.\overline{(\bar{C} + \bar{D})\bar{B}},$$

sea:

$$X = \bar{A}.(B + \bar{C}).(\overline{\bar{C} + \bar{D}} + B),$$

o:

$$X = \bar{A}.(B + \bar{C}).(CD + B) = \bar{A}B.$$

Obtenemos el mismo resultado: El resultado es independiente del orden en que se realicen las inversiones.

5. *Algunos circuitos lógicos importantes: O exclusivo, NO-O y NO-Y*

Los tres circuitos lógicos Y, O, inversores estudiados anteriormente representan los circuitos lógicos fundamentales. Con la ayuda de estos tres circuitos, se pueden realizar todas las funciones "booleanas" que puedan imaginarse. Podemos imaginar un calculador que se realizaría únicamente con circuitos O, Y, e inversores asociados entre ellos.

Por razones de comodidad y de facilidad de realización se han desarrollado circuitos que realizan otras funciones lógicas, en particular los circuitos:

O exclusivo.
NO - O o $\overline{O}$ (NO)
NO - Y o $\overline{Y}$ (NOY)
Inhibición.
Implicación

Los circuitos NO-O y NO-Y se conocen en la literatura anglosajona bajo el nombre de circuitos NOR y NAND (contracción de NOT-OR y NOT-AND). Estos circuitos no son circuitos fundamentales, es decir es posible para los dos últimos

realizar los circuitos O, Y e inversores utilizando únicamente circuitos NOR o circuitos NAND. El circuito O exclusivo es muy importante. Se utilizan ampliamente en la unidad aritmética, donde realiza la suma o diferencia sin resto de dos dígitos, así como los circuitos de generación y verificación de la paridad de una palabra que pasa a través del ordenador. Volveremos sobre este punto más adelante.

5.1 Función O exclusivo

Una función O exclusivo es una función de dos variables "booleanas" y *solamente dos*, de forma que el resultado es 1 y si una O la otra de las variables, *sólo una de ellas,* es igual a 1. El O exclusivo entre dos variables A y B se designa por $A \oplus B$ (leer A o exclusivo B). La tabla de verdad es la siguiente:

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Para leer esta tabla se procederá de la siguiente forma:

Referenciar las líneas en que la función a estudiar tiene el valor 1 (en nuestro caso la segunda y tercera línea). Se atribuye a continuación a A y B los valores $\overline{A}$ y $\overline{B}$ si $A = 0$ y $B = 0$ y A y B si $A = 1$ y $B = 1$. A continuación se hace la suma de todas las líneas en que la función es 1. Obtenemos así:

$$A \oplus B = \overline{A}B + A\overline{B}\,. \tag{28}$$

Si leemos los ceros en la columna $A \oplus B$ obtenemos la inversa de la función, sea:

$$\overline{A \oplus B} = \overline{A}\,\overline{B} + AB\,. \tag{29}$$

Estas dos últimas expresiones son muy importantes y de uso muy frecuente en el álgebra de Boole. Aconsejamos al lector que las retenga. Pueden obtenerse otras expresiones de $A \oplus B$. Así, de (29):

$$A \oplus B = \overline{\overline{A}\,\overline{B} + AB} \tag{30}$$

es decir, aplicando el teorema de De Morgan:

$$A \oplus B = (A + B)(\overline{A} + \overline{B}) \tag{31}$$

$$A \oplus B = (A + B)\,\overline{AB}\,. \tag{32}$$

Podemos realizar un circuito O exclusivo con los tres esquemas lógicos siguientes que corresponden respectivamente a (28), (31) y (32) (Figs. 17, 18 y 19).

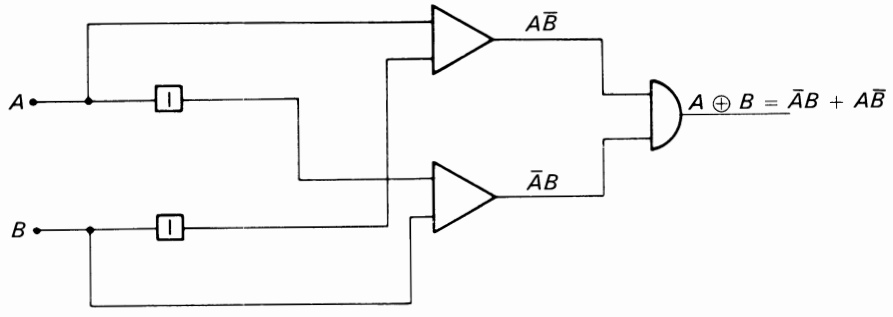


Fig. 17. *Circuito* O *exclusivo realizado siguiendo* (28).

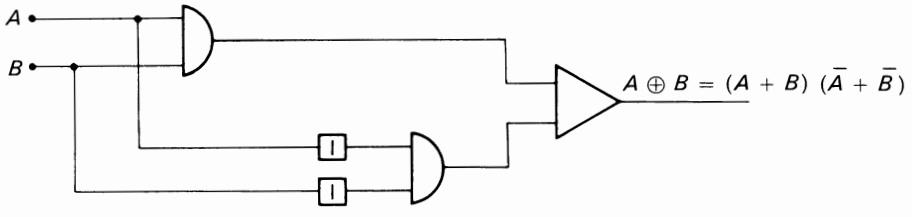


Fig. 18. *Circuito* O *exclusivo realizado siguiendo* (31).

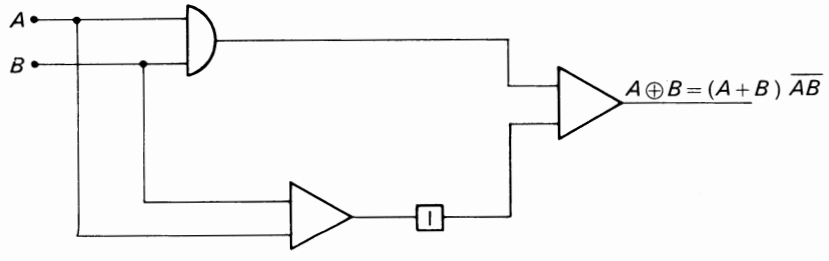


Fig. 19. *Circuito* O *exclusivo realizado siguiendo* (32).

Se observará que el circuito de la figura 19 es el más económico de los tres. Es éste el que se utilizará preferentemente. El símbolo general de un circuito O exclusivo es el de la figura 20. En este modo de representación no se tiene en cuenta la estructura interna del circuito. Todo lo que se sabe es que si se aplican

las dos variables "booleanas" A y B en las dos entradas se obtendrá a la salida la función $A \oplus B$.

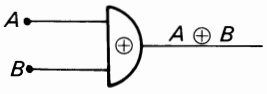


Fig. 20. *Símbolo de un circuito* O *exclusivo.*

5.2 Algunos teoremas importantes relativos al circuito O exclusivo

El circuito *O exclusivo* es uno de los circuitos más importantes de las calculadoras. Permite controlar la paridad (o imparidad) de las palabras en memoria así como la adición o la sustracción de dos o más variables "booleanas". A continuación damos algunos teoremas y fórmulas importantes relativas al circuito O exclusivo. Hemos demostrado en el párrafo anterior las siguientes relaciones de definición:

$$\begin{aligned} 0 \oplus 0 &= 0 \\ 0 \oplus 1 &= 1 \\ 1 \oplus 0 &= 1 \\ 1 \oplus 1 &= 0 \,. \end{aligned} \tag{33}$$

Estas fórmulas pueden generalizarse. Según la definición: $A \oplus B = \overline{A}B + A\overline{B}$, obtendremos:

$$\begin{aligned} A \oplus 0 &= A \qquad A \oplus A = 0 \\ A \oplus 1 &= \overline{A} \qquad A \oplus \overline{A} = 1 \,. \end{aligned} \tag{34}$$

La operación O exclusivo es conmutativa:

$$A \oplus B = B \oplus A \,. \tag{35}$$

Es suficiente sustituir A por B y recíprocamente, en la fórmula de definición. La operación O exclusivo es asociativa:

$$A \oplus B \oplus C = A \oplus (B \oplus C) = (A \oplus B) \oplus C \,. \tag{36}$$

Tenemos en efecto:

$$A \oplus B \oplus C = (\overline{A \oplus B})\, C + (A \oplus B)\, \overline{C} = (\overline{A}\overline{B} + AB)\, C + (\overline{A}B + A\overline{B})\, \overline{C} \,,$$

$$(B \oplus C) \oplus A = (\overline{B \oplus C})\, A + (B \oplus C)\, \overline{A} = (\overline{B}\overline{C} + BC)\, A + (\overline{B}C + B\overline{C})\, A \,.$$

Estas dos expresiones son idénticas. La operación o exclusivo verifica la distributividad:

$$A(B \oplus C) = AB \oplus AC \,. \tag{37}$$

En efecto, podemos escribir:

$$AB \oplus AC = \overline{AB}AC + AB\overline{AC} = (\overline{A} + \overline{B})\,AC + AB(\overline{A} + \overline{C}),$$
$$AB \oplus AC = A\overline{B}C + AB\overline{C} = A(\overline{B}C + B\overline{C}),$$
$$AB \oplus AC = A(B \oplus C).$$

Otras fórmulas interesantes:

Utilizando la fórmula de definición se demostrará análogamente:

$$\begin{aligned}
&AB \oplus \overline{A}C = AB + \overline{A}C \\
&A \oplus AB = A\overline{B} \\
&A \oplus \overline{A}B = A + B \\
&A \oplus (A + B) = \overline{A}B \\
&A \oplus (\overline{A} + B) = \overline{A}\overline{B} \\
&A \oplus B \oplus AB = A + B \\
&\overline{A \oplus B} = \overline{A} \oplus B = A \oplus \overline{B}.
\end{aligned} \tag{38}$$

Teorema 1

Si la operación o exclusivo se aplica a un número de variables idénticas, el resultado es nulo si el número de variables es par, e igual a A si el número de variables idénticas es impar:

$$A \oplus A \oplus A \cdots \oplus A = 0 \tag{39}$$

si el número de variables A es par.

$$A \oplus A \oplus A \cdots A = A \tag{40}$$

si el número de variables A es impar.

Ejemplo 1. *Calculemos la siguiente función:*

$$X = A \oplus A \oplus A \oplus A \oplus A.$$

El número de variables *idénticas A* es impar. Calculemos por separado el resultado de cada circuito o exclusivo.

$$X = \underbrace{A \oplus A}_{0} \oplus \underbrace{A \oplus A}_{0} \oplus A$$
$$= \underbrace{0 \oplus 0}_{0} \oplus A$$
$$= \underbrace{0 \oplus A}_{A}$$

Obtenemos que $X = A$.

Ejemplo 2. *Sea calcular la siguiente función:*

$$X = A \oplus A \oplus A \oplus A .$$

El número de variables idénticas es par:

$$X = \underbrace{\underbrace{A \oplus A}_{0} \oplus \underbrace{A \oplus A}_{0}}_{0}$$

Se obtiene $X = 0$.

En lugar de proceder por grupos de dos variables se las puede leer sucesivamente modificando el resultado a cada nueva operación. Así tendremos, para el primer ejemplo:

Ejemplo 1a:

$$A \oplus A = 0$$
$$A \oplus A \oplus A = 0 \oplus A = A$$
$$A \oplus A \oplus A \oplus A = A \oplus A = 0$$
$$X = A \oplus A \oplus A \oplus A \oplus A = 0 \oplus A = A .$$

Siguiendo este ejemplo para cuatro variables idénticas, $X = 0$. Obteniéndose el resultado de los ejemplos 1 y 2 anteriores.

Teorema 2

Si en una sucesión de operaciones o exclusivo el número de variables que tiene el valor 1 es par (las otras variables son cero) el resultado es cero. El resultado es 1 si el número de variables que tiene el valor 1 es impar:

$$A \oplus B \oplus C \oplus \cdots = 0 \tag{41}$$

si un número par de variables tiene el valor 1.

$$A \oplus B \oplus C \oplus \cdots = 1 \tag{42}$$

si un número impar de variables tiene el valor 1.

Ejemplo:

$$X = 1 \oplus 0 \oplus 1 \oplus 1 \oplus = 1$$
$$X = 0 \oplus 1 \oplus 1 \oplus 0 = 0 .$$

Teorema 3

Si una sucesión de operaciones o exclusivo:

$$AB = \cdots AC = \cdots BC = 0 ,$$

el resultado es la suma de las variables:

$$A \oplus B \cdots \oplus C = A + B + \cdots + C . \tag{43}$$

Ejemplo. *Calcular* $A \oplus B \oplus C$ *con las condiciones* $AB = AC = BC = 0$.

Tenemos:

$$A \oplus B = (A + B)\,\overline{AB} .$$

Si $AB = 0$:

$$A \oplus B = A + B .$$

En estas condiciones:

$$A \oplus B \oplus C = (A + B + C)\,\overline{[(A + B)\,C]} .$$

La cantidad entre corchetes es igual a 1 ($AC = BC = 0$). Se verifica que:

$$A \oplus B \oplus C = A + B + C .$$

5.3 Circuito de coincidencia

El circuito que realiza la función inversa del circuito o exclusivo, es decir proporcionando un 1 en salida cuando las dos entradas son idénticas (00 ó 11 simultáneamente en las dos entradas) se llama circuito de coincidencia. Realiza la función.

$$A \odot B = \overline{A \oplus B} , \tag{44}$$

que responde a la siguiente tabla de verdad.

A	B	$A \oplus B$	$A \odot B$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

Leyendo la tabla obtenemos:

$$\begin{aligned} A \odot B &= \overline{A}\,\overline{B} + AB , \\ A \odot B &= (A + \overline{B})\,(\overline{A} + B) . \end{aligned} \tag{45}$$

La primera de estas expresiones corresponde al valor 1, la segunda al valor 0 de la función. Las fórmulas (45) pueden transformarse igualmente mediante el teorema de De Morgan:

$$\begin{aligned} A \odot B &= (A + \overline{B})\,\overline{A\overline{B}} \\ A \odot B &= (\overline{A} + B)\,\overline{\overline{A}B} . \end{aligned} \tag{46}$$

La expresión $A \odot B$ se lee "A coincidencia con B". La función $A \odot B$ se define mediante las siguientes operaciones:

$$\begin{aligned} 0 \odot 0 &= 1 , \\ 1 \odot 0 &= 0 , \\ 0 \odot 1 &= 0 , \\ 1 \odot 1 &= 1 , \end{aligned} \tag{47}$$

y de una forma más general si A es una variable "booleana" se verifica mediante la tabla de verdad las siguientes expresiones:

$$\begin{aligned} A \odot 0 &= \overline{A} \\ A \odot 1 &= A \\ A \odot A &= 1 \\ A \odot \overline{A} &= 0 . \end{aligned} \tag{48}$$

Estas expresiones se obtienen invirtiendo los dos números de (33) y (34). Se establecerán para el circuito de coincidencia fórmulas similares a las obtenidas para el circuito o exclusivo (Tabla 1). Se hace notar que la distributividad no se verifica para el circuito de coincidencia.

$$AB \odot AC \neq A(B \odot C) . \tag{49}$$

Igualmente se verifica que:

$$\begin{aligned} &(A + B) \odot (A + C) = A + B \odot C \\ &A \odot B = \overline{A \oplus B} = \overline{A}\overline{B} + AB \\ &\overline{A \odot B} = A \oplus B = \overline{A} \odot B = A \odot \overline{B} \\ &A \odot AB = A\overline{\overline{B}} \\ &A \odot \overline{A}B = \overline{A + B} \\ &A \odot (\overline{A} + B) = AB \\ &A \odot (A + B) = A + \overline{B} \\ &A \odot (A + \overline{B}) = A + B \\ &A(\overline{A} \odot B) = A\overline{B} \\ &\overline{A}(\overline{A} \odot B) = \overline{A}B \end{aligned} \tag{50}$$

Se pone de manifiesto en esta tabla que $A \odot B \odot AB = A + B$, es decir que

$$\overline{A \oplus B \oplus AB} \neq A \odot B \odot AB .$$

Tabla I. *Semejanzas destacables de las operaciones "o exclusivo" y "coincidencia"*

⊙ exclusivo $A \oplus B$	Coincidencia $A \odot B$
$A \oplus 0 = A$ $A \oplus 1 = \overline{A}$ $A \oplus A = 0$ $A \oplus \overline{A} = 1$	$A \odot 0 = \overline{A}$ $A \odot 1 = A$ $A \odot A = 1$ $A \odot \overline{A} = 0$
$A \oplus A \oplus \cdots \oplus A = 0$	$A \odot A \odot A \cdots \odot A = 1$
si el número A es par.	si el número de A es par.
$A \oplus A \oplus \cdots \oplus A = A$	$A \odot A \odot \cdots \odot A = A$
si el número A es impar	si el número de A es impar.
$A \oplus B \oplus C \cdots = 0$	$A \odot B \odot C \cdots = 1$
si un número par de variables tiene el valor 1.	si un número par de variables tiene el valor 0.
$A \oplus B \oplus C \cdots = 1$	$A \odot B \odot C \cdots = 0$
si un número impar de variables tiene el valor 1.	si un número impar de variables tiene el valor 1.
$A \oplus B = B \oplus A$	$A \odot B = B \odot A$
$A \oplus B \oplus C = (A \oplus B) \oplus C = A \oplus (B \oplus C)$	$A \odot B \odot C = (A \odot B) \odot C = A \odot (B \odot C)$
$AB \oplus AC = A(B \oplus C)$	$(A+B) \odot (A+C) = A + (B \odot C)$
$A \oplus B \oplus AB = A + B$	$A \odot B \odot AB = A + B$
$A \oplus B \oplus \cdots \oplus C = A + B + C$ si $AB = \cdots AC = \cdots BC = 0$.	$A \odot B \odot \cdots \odot C = A + B + \cdots C$ si $AB = \cdots AC = \cdots BC = 1$.
$AB \oplus \overline{A}C = AB + \overline{A}C$	$AB \odot \overline{A}C = \overline{AB + \overline{A}C}$
$A \oplus B = A\overline{B} + \overline{A}B = (A+B)(\overline{A}+\overline{B})$	$A \odot B = \overline{A}\overline{B} + AB = (\overline{A}+B)(A+\overline{B})$

En efecto, se tiene

$$A \oplus B \oplus AB = A \odot B \odot AB = A + B . \tag{51}$$

Realización del circuito de coincidencia

Existen cinco formas de realizar el circuito de coincidencia:

- realizar los esquemas lógicos de las ecuaciones (45) y (46),
- invertir la salida del circuito O exclusivo.

Estos esquemas se dan en la figura 21.

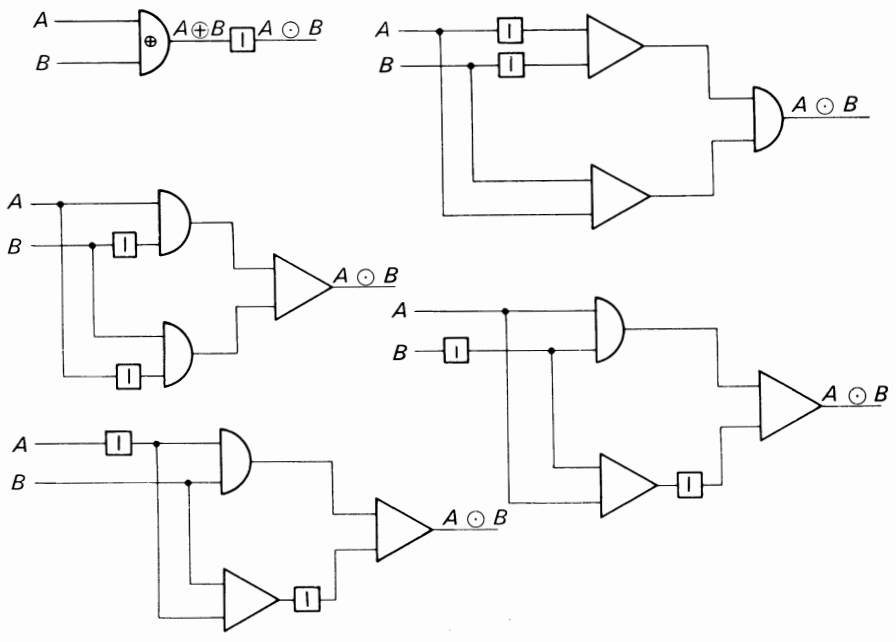


Fig. 21. *Cinco formas de realizar la coincidencia* $A \odot B$.

5.3.1 *Inversores realizados con un circuito o exclusivo o un circuito de coincidencia*

Según (34), $A \oplus 1 = \bar{A}$ y, según (48), $A \odot 0 = \bar{A} \odot$. Disponemos así de un medio de realizar un inversor. Es suficiente mantener una entrada del circuito O exclusivo con 1, o una entrada del circuito de coincidencia con 0.

5.4 Aplicación del circuito O exclusivo: circuitos generadores de paridad y de control de paridad

El circuito O exclusivo se adapta, particularmente bien, al control de paridad de los datos tratados por el ordenador.

Por paridad de una palabra se entiende el número de 1 que contiene esa palabra, si es par se hablará de paridad par y si es impar de paridad impar (imparidad). Un medio para controlar el tratamiento de las palabras de memoria por el ordenador es conseguir que todas éstas posean antes del tratamiento un *número* par de 1.

Se necesita prever circuitos que generen un 1 suplementario para las palabras que contengan un número impar de 1 en el momento de su introducción en memoria.

Recordemos que se entiende por palabra tanto las instrucciones como los datos. Durante el tratamiento se comprobará que se respeta la paridad, en los distin-

tos componentes de la máquina. El circuito que genera un bit de paridad está constituido por una pirámide de circuitos O exclusivos. Así, sea el número 1011 1100, que contiene un número impar de 1. Se compara mediante los circuitos O exclusivos todos los bits de la palabra tomados dos a dos (Fig. 21). Las salidas de cada circuito O exclusivo se comparan nuevamente dos a dos mediante la utilización de circuitos O exclusivos, así hasta obtener una salida única. Esta será 1 si el número de 1 en la palabra es impar. Esta salida se utilizará para, junto con 1, hacerla pasar a través de un elemento "paridad" que se asocia a la palabra estudiada. Este constituye lo que se llama un bit de paridad. El esquema del circuito es el de la figura 22.

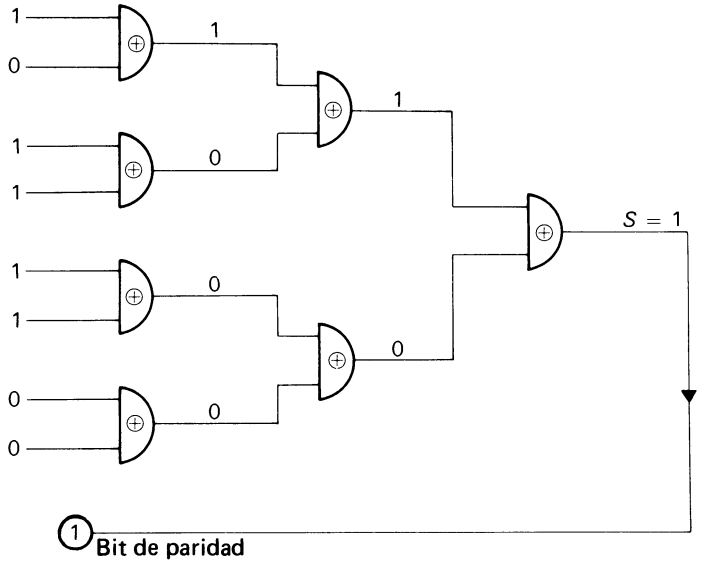


Fig. 22. *Circuito generador de un bit de paridad.*

Así se asocia a todas las palabras, desde su introducción en memoria, un bit de paridad. Un circuito idéntico al precedente controla seguidamente esta palabra en diferentes puntos de la máquina durante el funcionamiento. Cuando la salida sea 1 se detectará un error, es decir cuando el número de 1 en la palabra sea impar (Fig. 23).

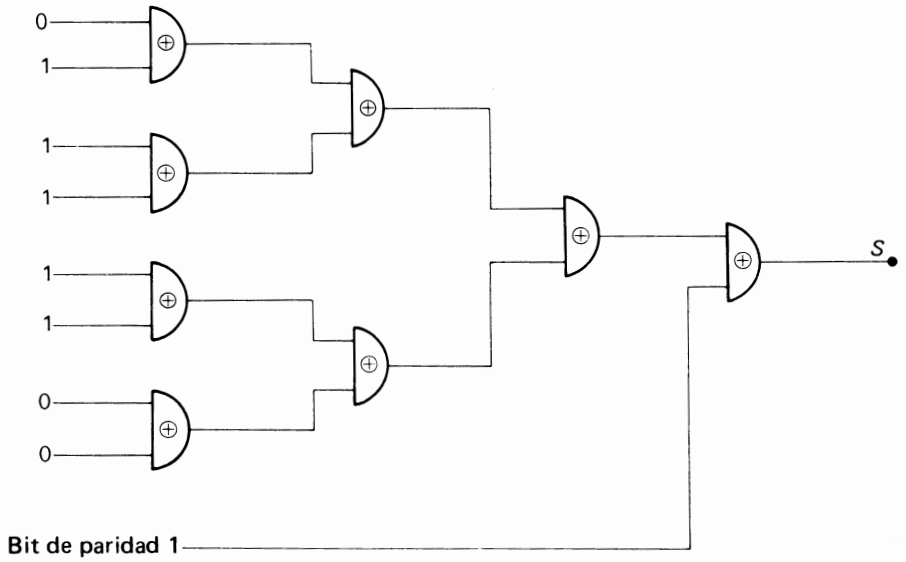


Fig. 23. *Circuito de control de la paridad (S = 0, sin error; S = 1, error).*

Estos esquemas pueden establecerse analíticamente. En efecto, en el ejemplo escogido se desea que cada palabra contenga un número par de 1, llamemos x_n, x_{n-1}, ..., x_1, x_0 a los dígitos de cada palabra en memoria. Sea p el dígito de paridad, tendremos, de acuerdo con el teorema 2 (41):

$$x_n \oplus x_{n-1} \oplus \cdots \oplus x_1 \oplus x_0 \oplus p = 0 \,. \tag{52}$$

En efecto, tenemos siempre un número par de variables que tienen el valor 1. De la expresión anterior obtenemos:

$$x_n \oplus x_n \oplus x_{n-1} \oplus x_{n-1} \cdots x_1 \oplus x_1 \oplus x_0 \oplus x_0 \oplus p = 0 \oplus x_n \oplus x_{n-1} \oplus \cdots x_1 \oplus x_0 \,,$$

es decir:

$$p = x_n \oplus x_{n-1} \cdots \oplus x_1 \oplus x_0 \tag{53}$$

que determina el circuito generador del bit de paridad.

Ejemplo. Consideremos un número de cuatro dígitos x_3, x_2, x_1 y x_0; tendremos:

$$p = x_3 \oplus x_2 \oplus x_1 \oplus x_0 \,,$$

y el siguiente circuito (Fig. 24).

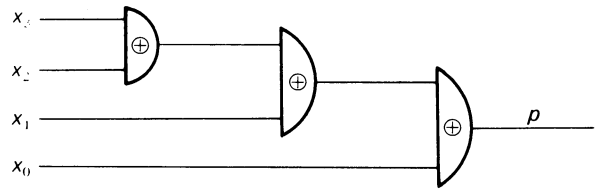


Fig. 24. *Generación de un bit de paridad en una palabra de 4 dígitos (segunda versión).*

El circuito de control se determinará mediante (52). La salida S será nula si no hay ceros (número par de 1 en la palabra) ó 1 si hay ceros (número impar de 1 en la palabra), figura 25.

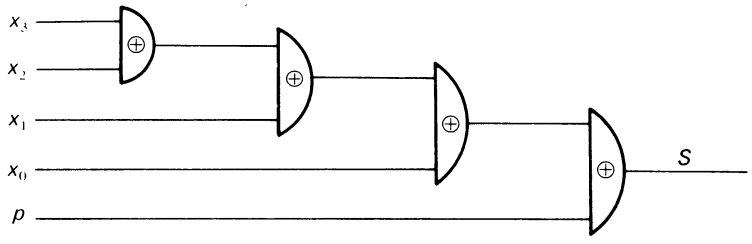


Fig. 25. *Control de la paridad en una palabra de cuatro dígitos (segunda versión).*

Los esquemas de las figuras 24 y 25 son idénticos a los esquemas de las figuras 22 y 23. En un caso los circuitos o exclusivos se asocian dos a dos, mientras que en el otro, los circuitos o exclusivos se disponen en cadena. En los dos casos, el resultado a la salida es el mismo (ver ejemplos 1 y 2, en p. 23 y 24).

5.5 Función NOR

Una función NOR es una operación lógica efectuada sobre dos o más variables "booleanas" de forma que el resultado es 1 si, y solamente si, una y la otra o la totalidad de las variables es cero. La operación NOR se designa mediante $A \downarrow B$ y se expresa por "A Pierce B" del nombre de Pierce que fue el primero en utilizar esta notación. La tabla de verdad limitada a dos variables se da en la figura 26.

A	B	$A \downarrow B$
0	0	1
0	1	0
1	0	0
1	1	0

Fig. 26. *Tabla de verdad de la función NOR A ↓ B.*

La operación NOR se define mediante:

$$\begin{aligned} 0 \downarrow 0 &= 1 \\ 0 \downarrow 1 &= 0 \\ 1 \downarrow 0 &= 0 \\ 1 \downarrow 1 &= 0 , \end{aligned} \tag{54}$$

y de una forma más general si A es una variable booleana:

$$\begin{aligned} A \downarrow 0 &= \overline{A} \\ A \downarrow 1 &= 0 \\ A \downarrow A &= \overline{A} \\ A \downarrow \overline{A} &= 0 . \end{aligned} \tag{55}$$

La función NOR se expresará mediante la tabla de verdad por:

$$A \downarrow B = \overline{A}\,\overline{B} = \overline{A + B} \tag{56}$$

o:

$$A \downarrow B = (A + \overline{B})(\overline{A} + B)(\overline{A} + \overline{B}) . \tag{57}$$

Esta última expresión corresponde al valor 0 de la función NOR. La expresión (56) da los dos esquemas equivalentes de la figura 27.

Fig. 27. *Esquemas equivalentes de la función* NOR.

Se observará que si sólo se dispone de dos valores ciertos A y B, la segunda versión, que sólo comprende un inverso, es más económica que la primera. La función NOR puede generalizarse a n variables. En este caso tenemos n entradas, figura 28.

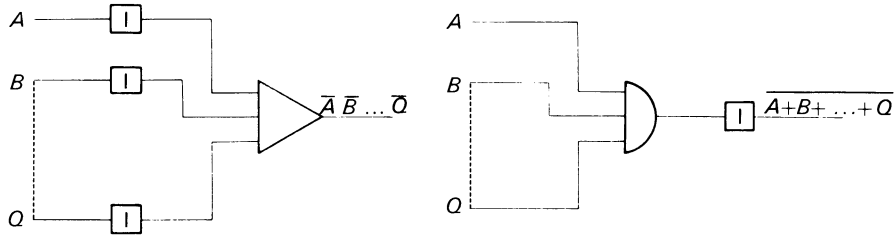


Fig. 28. *Generalización de la función* $\overline{O}$ *para n entradas.*

El circuito NOR que es la traducción anglosajona de NO O, se representa simbólicamente por un rectángulo que contiene el término NOR (Fig. 29).

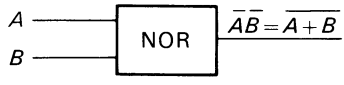


Fig. 29. *Símbolo de un circuito* NOR ($\overline{O}$).

La función NOR no es una operación lógica fundamental, es posible reconstituir todas las operaciones lógicas fundamentales, inversión, adición y multiplicación lógica con uno o varios circuitos NOR. Así en la expresión:

$$X = \overline{A + B} \tag{58}$$

que representa la salida X de un circuito NOR con dos entradas A y B, $A = B$, tenemos:

$$X = \overline{A}\,. \tag{59}$$

Hemos realizado el equivalente de un inversor. Es suficiente aplicar la variable única A a las dos entradas de un circuito NOR para obtener $\overline{A}$ a la salida. El mismo resultado se puede obtener haciendo $B = 0$ (Fig. 30).

Fig. 30. *Inversión mediante un circuito* NOR.

Igualmente obtenemos un circuito O mediante dos circuitos NOR dispuestos en cadena. Podemos escribir:

$$X = A + B = \overline{\overline{A + B}} = (A \downarrow B) \downarrow (A \downarrow B)\,. \tag{60}$$

Es suficiente disponer de un primer circuito NOR con dos entradas cuya salida entrega $\overline{A + B}$ seguido de un inversor realizado en un circuito NOR de acuerdo con la figura 30 para obtener $A + B$, es decir el equivalente de un circuito O, figura 31. Este circuito lo podemos generalizar para n entradas. Un circuito Y podrá obtenerse igualmente a partir de circuitos NOR. En efecto, de acuerdo con el teorema de De Morgan, podemos escribir:

$$X = AB = \overline{\overline{A} + \overline{B}} = (A \downarrow A) \downarrow (B \downarrow B) . \tag{61}$$

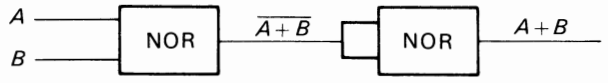


Fig. 31. *Circuito* O *realizado por el circuito* NOR.

Esta última expresión es el NOR de A y B. Es suficiente disponer a la entrada dos inversores de acuerdo con la figura 30 para obtener el circuito Y, figura 32. Este circuito puede generalizarse a n entradas.

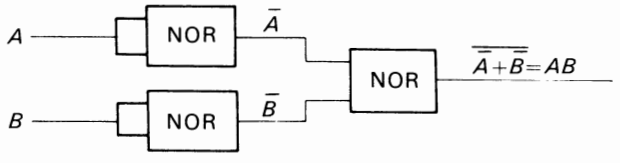


Fig. 32. *Circuito* Y *realizado con tres circuitos* NOR.

Examinemos ahora la realización de un circuito O exclusivo y de un circuito de coincidencia con la ayuda de un circuito NOR. El esquema de la figura 33 proporciona en la salida $A \odot B$ o $A \oplus B$. Este circuito se utilizará posteriormente en los adicionadores con circuitos NOR.

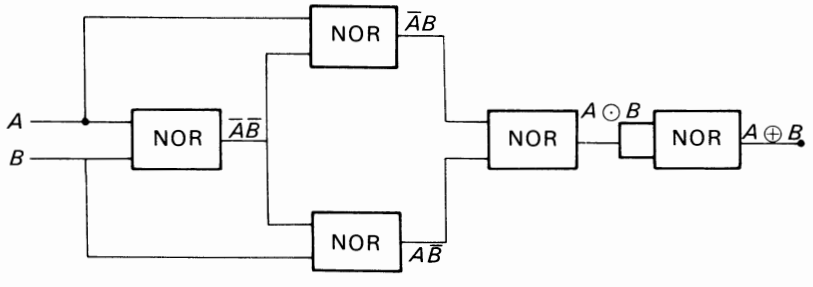


Fig. 33. *Realización de un circuito de coincidencias* $A \odot B$ *y de un* O *exclusivo* $A \oplus B$ *con la ayuda de circuitos* NOR.

5.6 Función NAND

Una función NAND es una operación lógica que se efectúa sobre dos (o varias) variables "booleanas", de tal forma que el resultado es 0 si y solamente si las dos variables o la totalidad de las variables son 1. La operación NAND limitada a dos variables, A y B, se designa por A/B y se expresa mediante "A por B". La tabla de verdad es la de la figura 34.

A	B	A/B
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 34. *Tabla de verdad de la función* NAND : *A/B.*

La operación NAND se define mediante:

$$\begin{aligned} 0/0 &= 1 \\ 0/1 &= 1 \\ 1/0 &= 1 \\ 1/1 &= 0 \end{aligned} \tag{62}$$

y de una forma más general si A es una variable "booleana" mediante:

$$\begin{aligned} A/0 &= 1 \\ A/1 &= \overline{A} \\ A/A &= \overline{A} \\ A/\overline{A} &= 1\ . \end{aligned} \tag{63}$$

La lectura de la tabla de verdad da para la función NAND las siguientes expresiones:

$$A/B = \overline{A}\,\overline{B} + \overline{A}B + A\overline{B} \tag{64}$$

$$A/B = \overline{A} + \overline{B} = \overline{AB}\ . \tag{65}$$

La primera se simplifica y se reduce a (65). El circuito NAND puede representarse mediante un circuito Y seguido de un inversor (por lo que se llaman $\overline{\text{Y}}$) o por un circuito O con dos entradas $\overline{A}$ y $\overline{B}$ (Fig. 35).

Fig. 35. *Esquemas equivalentes de la función* $\overline{Y}$ (NAND).

Si sólo se dispone de dos valores A y B hay que hacer notar que el circuito NAND realizado con un circuito Y es más económico que su equivalente realizado con un circuito O. El circuito puede generalizarse para n entradas (Fig. 36).

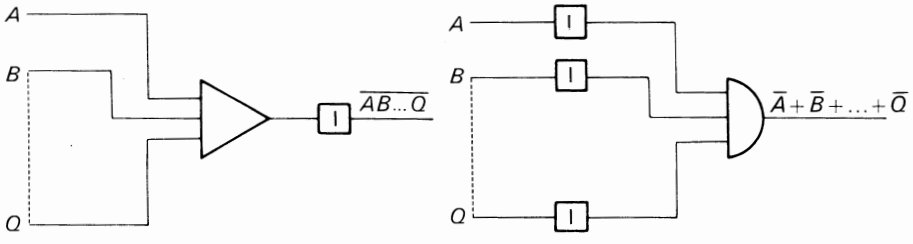


Fig. 36. *Generalización de la función* $\overline{Y}$ (NAND) *para n entradas.*

El símbolo de un circuito NAND es un rectángulo que lleva la inscripción NAND (Fig. 37).

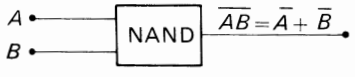


Fig. 37. *Símbolo de un circuito* NAND ($\overline{Y}$).

La función NAND no es una función lógica elemental. Es posible obtener las funciones lógicas elementales inversión, suma y multiplicación lógica, mediante una combinación de circuitos NAND. Si en la expresión:

$$X = \overline{AB} = \overline{A} + \overline{B} = A/B \tag{66}$$

hacemos $B = A$, obtenemos

$$X = \overline{A}\,. \tag{67}$$

Se realiza así un inversor con un circuito NAND. Es suficiente aplicar la variable A a las dos entradas (Fig. 38). Hay varias posibilidades para realizar una inversión. Se puede utilizar:

- un inversor,
- un circuito NOR,
- un circuito NAND,
- un circuito O exclusivo o un circuito de coincidencia.

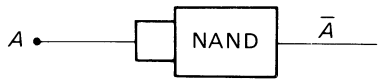


Fig. 38. *Inversión mediante un circuito* NAND.

Un circuito O realiza la función:

$$X = A + B = \overline{\bar{A}\bar{B}} = (A/A)/(B/B) . \tag{68}$$

Es suficiente el disponer las variables A y B a la entrada de un circuito NAND y reunir las salidas a un circuito NAND utilizado como inversor (Fig. 39).

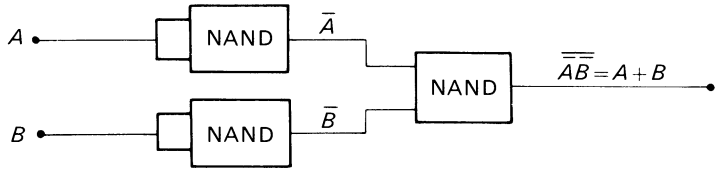


Fig. 39. *Circuito* O *realizado con tres circuitos* NAND.

Igualmente podemos escribir:

$$X = AB = \overline{\overline{AB}} = (A/B)/(A/B) . \tag{69}$$

Así un circuito Y puede realizarse mediante un circuito NAND ($\overline{AB}$) seguido de un inversor (Fig. 40).

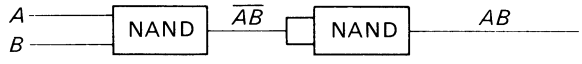


Fig. 40. *Circuito* Y *realizado con dos circuitos* NAND.

También podemos realizar un circuito O exclusivo o un circuito de coincidencia, utilizando únicamente circuitos NAND (Fig. 41).

Teniendo en cuenta la figura 33, se verá la simetría de las dos realizaciones.

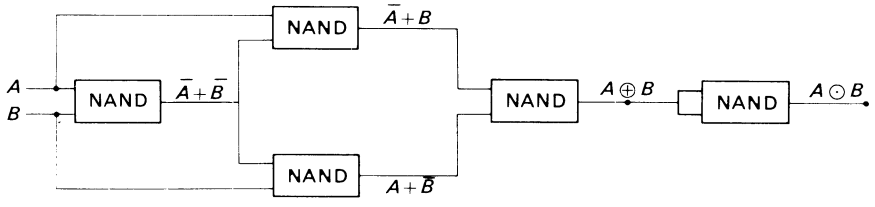


Fig. 41. *Realización de un circuito* O *exclusivo* $A \oplus B$ *y de un circuito de coincidencia* $A \odot B$ *mediante circuitos* NAND.

5.7 Paso de un circuito NOR a un circuito NAND

De acuerdo con lo que acabamos de ver, para pasar de un circuito NOR a un circuito NAND es suficiente cambiar los productos por sumas. Recíprocamente, para pasar de un circuito NAND a un circuito NOR, se cambiarán las sumas por productos. Así un esquema que utiliza circuitos NOR produce a la salida la función:

$$X = (AB + CD)\,EF\,,$$

este mismo circuito en el cual se sustituyen los circuitos NOR por circuitos NAND producirá a la salida la función:

$$X = (A + B)(C + D) + E + F\,,$$

obtenido cambiando en la expresión anterior los productos en sumas y las sumas en productos.

5.7.1 *Realización de un circuito mediante circuitos* NOR *o de circuitos* NAND

Según lo que acabamos de ver, un circuito puede realizarse con circuitos NOR o circuitos NAND. Para una realización con los NOR la función lógica se expresará bajo la forma de un producto de sumas. Es suficiente entonces sustituir todos los símbolos lógicos por el símbolo NOR en la expresión de la función a realizar.

Ejemplo. *Sea realizar la función lógica:*

$$X = (A + B)\,(C + D)$$

con circuitos NOR.

Hemos visto anteriormente que:

$$A + B = \overline{A \downarrow B}$$
$$AB = \bar{A} \downarrow \bar{B}\,.$$

De donde, sustituyendo en X:

$$X = (\overline{A \downarrow B})\,(\overline{C \downarrow D})\,,$$

o:

$$X = (\overline{\overline{A \downarrow B}}) \downarrow (\overline{\overline{C \downarrow D}})\,.$$

es decir

$$X = (A \downarrow B) \downarrow (C \downarrow D)\,.$$

Comparando esta expresión con su valor de partida se observará que todos los símbolos lógicos se han sustituido por el símbolo NOR. Si se desea expresar la función X con circuitos NAND, se expresará ésta bajo la forma de una suma de productos. Es suficiente, en este caso, sustituir los símbolos lógicos por el símbolo NAND.

Ejemplo. *Sea realizar la función* $X = AB + CD$ *con los circuitos* NAND.

Según lo visto anteriormente:

$$AB = \overline{A/B},$$
$$A + B = \bar{A}/\bar{B},$$

De donde llevando estos resultados a X:

$$X = \overline{A/B} + \overline{C/D},$$

ó:

$$X = (\overline{\overline{A/B}})/(\overline{\overline{C/D}}),$$

es decir

$$X = (A/B)/(C/D).$$

Comparando esta expresión con la expresión de origen se observará que todos los símbolos lógicos se han sustituido por el símbolo NAND.

5.8 Circuitos de inhibición

Otros dos circuitos lógicos se utilizan igualmente: los circuitos de inhibición y de implicación. No obstante estos circuitos se utilizarán menos que los anteriores. A continuación damos su definición. Se llama inhibición una operación lógica realizada sobre dos variables "booleanas" A y B de tal forma que la salida es 1 sólo y únicamente si $A = 1$ y $B = 0$.

De una forma más general se obtiene la variable A a la salida (0 ó 1) si no está presente B ($B = 0$). La figura 42 da la tabla de verdad de la operación de inhibición.

A	B	$A\bar{B}$
0	0	0
0	1	0
1	0	1
1	1	0

Fig. 42. *Tabla de verdad de la operación de inhibición.*

La operación de inhibición realiza las siguientes operaciones:

$$\begin{aligned} 0.\bar{0} &= 0 \\ 0.\bar{1} &= 0 \\ 1.\bar{0} &= 1 \\ 1.\bar{1} &= 0, \end{aligned} \qquad (70)$$

y de una forma más general, si A es una variable "booleana":

$$\begin{aligned} 0.\overline{A} &= 0 \\ 1.\overline{A} &= \overline{A} \\ \overline{A}.\overline{A} &= \overline{A} \\ A.\overline{A} &= 0\ . \end{aligned} \tag{71}$$

La definición del circuito de *inhibición* es $A\overline{B}$. Su realización está materializada en la figura 43. Se llama igualmente circuito "A no B".

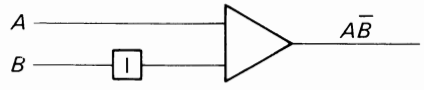


Fig. 43. *Circuito de inhibición.*

5.9 Implicación

Se llama implicación de dos variables "booleanas" A y B la función $A + \overline{B}$. Es la inversa de la función inhibición B no A, es decir $\overline{A}B$. En efecto, la tabla de verdad es:

$$A + \overline{B} = \overline{\overline{A}B} \tag{72}$$

da la figura 44.

A	B	$A + \overline{B}$
0	0	1
0	1	0
1	0	1
1	1	1

Fig. 44. *Tabla de verdad de la función implícita* $A + \overline{B}$.

Según esta tabla, tenemos:

$$\begin{aligned} 0 + \overline{0} &= 1 \\ 0 + \overline{1} &= 0 \\ 1 + \overline{0} &= 1 \\ 1 + \overline{1} &= 1\ , \end{aligned} \tag{73}$$

y de una forma más general, si A es una variable "booleana":

$$\begin{aligned} 0 + \overline{A} &= \overline{A} \\ \overline{A} + 1 &= 1 \\ \overline{A} + \overline{A} &= \overline{A} \\ A + \overline{A} &= 1 \,. \end{aligned} \tag{74}$$

La función implicación se realiza mediante el esquema de la figura 45.

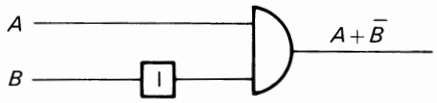


Fig. 45. *Realización de la operación implicación.*

Tabla recapitulativa

La tabla, figura 46, reúne las principales funciones lógicas estudiadas hasta aquí, así como sus propiedades. Se dedicará una atención especial al circuito O exclusivo y al circuito de coincidencia.

6. Funciones realizadas por dos variables «booleanas»

Consideremos dos variables "booleanas" A y B. La suma de las combinaciones da cuatro casos posibles. Para cada una de estas combinaciones el resultado de la función realizada con estas dos variables puede ser 0 ó 1. Hay $2^4 = 16$ funciones realizables. Estas funciones son las siguientes:

$$\begin{aligned} F_1 &= 0 & F_9 &= \overline{A}\,\overline{B} = \overline{A + B} \\ F_2 &= A & F_{10} &= \overline{AB} = \overline{A} + \overline{B} \\ F_3 &= B & F_{11} &= A + B \\ F_4 &= \overline{A} & F_{12} &= \overline{A} + B \\ F_5 &= \overline{B} & F_{13} &= A + \overline{B} \\ F_6 &= AB & F_{14} &= \overline{A}B + A\overline{B} \\ F_7 &= \overline{A}B & F_{15} &= \overline{\overline{A}B + A\overline{B}} = \overline{A}\,\overline{B} + AB \\ F_8 &= A\overline{B} & F_{16} &= 1 \,. \end{aligned} \tag{75}$$

Estas funciones representan la totalidad de las funciones lógicas que se pueden realizar con dos variables "booleanas" A y B.

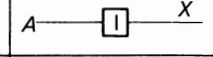
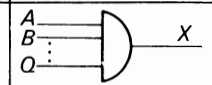
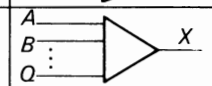
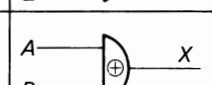
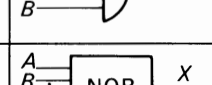
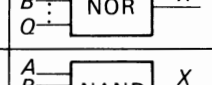
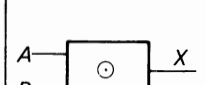
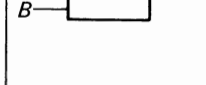
Función	Símbolo	Salida	Fórmulas fundamentales
Inversión		$X=\overline{A}$	$A+0=A \quad A.0=0$ $A+1=1 \quad A.1=A \quad \overline{\overline{A}}=A$ $A+A=A \quad A.A=A$ $A+\overline{A}=1 \quad A.\overline{A}=0$
Adición lógica		$X=A+B+\cdots+Q$	
Multiplicación lógica		$X=AB \ldots Q$	$A+AB=A$ $A+\overline{A}B=A+B$ $A(A+B)=A$ $(A+B)\ (A+\overline{B})=A$ $(A+B)\ (A+C)=A+BC$ $A+B=B+A$ Conmutatividad de la suma y del producto $AB=BA$ $A+(B+C)=(A+B)+C$ Asociación de la suma y del producto $A(BC)=(AB)\ C$ $A(B+C)=AB+AC$ Distributividad
O exclusivo		$X=\overline{A}B+A\overline{B}=A\oplus B$ $\overline{X}=\overline{A}\overline{B}+AB=A\odot B$ $X=(A+B)\ (\overline{A}+\overline{B})$	
NOR		$X=\overline{A+B+\cdots+Q}=\overline{A}\overline{B} \ldots \overline{Q}$	
NAND		$X=\overline{AB \ldots Q}=\overline{A}+\overline{B}+\cdots+\overline{Q}$	
Coincidencia		$X=\overline{A}\overline{B}+AB$ $X=A\odot B$ $X=\overline{A\oplus B}$ $X=(A+\overline{B})\ (\overline{A}+B)$ $X=(A+\overline{B})\ A\overline{\overline{B}}$ $X=(\overline{A}+B)\ \overline{\overline{A}}B$	
Inhibición		$X=A\overline{B}$	
Implicación		$X=A+\overline{B}$	

Fig. 46. *Tabla recapitulativa de las principales propiedades del álgebra de Boole*

6.1 Formas canónicas de una función "booleana"

Una función "booleana" de varias variables puede representarse *siempre* por dos expresiones equivalentes. Así, de acuerdo con lo que hemos visto en el estudio del circuito O exclusivo:

$$A \oplus B = \overline{A}B + A\overline{B} \tag{76}$$

o:

$$A \oplus B = (A + B)(\overline{A} + \overline{B}) . \tag{77}$$

La primera expresión corresponde a las combinaciones de las variables para las cuales $A \oplus B = 1$, la segunda las combinaciones para las cuales $A \oplus B = 0$. Estas dos expresiones son *idénticas.* En efecto,

$$A \oplus B = (A + B)(\overline{A} + \overline{B}) = \overline{A}B + A\overline{B} .$$

La primera expresión es una suma de los términos $\overline{A}B$ y $A\overline{B}$. Cada uno de estos términos representa uno de los productos que se pueden formar con las dos variables A y B. La segunda expresión es un producto de los términos $(A + B)$ y $(\overline{A} + \overline{B})$. Cada uno de estos términos representa una de las sumas que se pueden formar con dos variables A y B. De una forma más general una función de varias variables "booleanas" podrá representarse siempre bajo una de las siguientes formas:

$$F = P_0 + P_1 + P_2 + \cdots + P_n \tag{78}$$

$$F = S_0 S_1 S_2 \ldots S_n .$$

Los términos P son los términos productos y los términos S son los términos suma. La primera forma corresponde a lo que se llama la forma *canónica disyuntiva,* la segunda expresión a lo que se llama la forma *canónica conjuntiva.* En el caso de una función de una sola variable, estas dos expresiones se confunden. La tabla de la figura 47 da los términos P y S para una función de dos variables A y B.

A	B	Términos P	Términos S
0	0	$P_0 = \overline{A}\overline{B}$	$S_0 = A + B$
0	1	$P_1 = \overline{A}B$	$S_1 = A + \overline{B}$
1	0	$P_2 = A\overline{B}$	$S_2 = \overline{A} + B$
1	1	$P_3 = AB$	$S_3 = \overline{A} + \overline{B}$

Fig. 47. *Términos P y S de una función de dos variables, A y B.*

Los términos P son los productos de dos variables para las cuales $F = 1$, es decir tales que $P_0 = P_1 = P_2 = P_3 = 1$. Los términos S son los productos de dos

variables para las cuales $F = 0$, es decir tales que $S_0 = S_1 = S_2 = S_3 = 0$.
Se observará que:

$$P_i = \overline{S_i}$$
$$S_i = \overline{P_i}\,. \tag{79}$$

La tabla de la figura 48 da los términos P y los términos S para una función de tres variables, A, B y C. Se observará que una función de n variables "booleanas" comprende 2^n términos P y S.

A	B	C	P términos	Términos S
0	0	0	$P_0 = \overline{A}\,\overline{B}\,\overline{C}$	$S_0 = A + B + C$
0	0	1	$P_1 = \overline{A}\,\overline{B}C$	$S_1 = A + B + \overline{C}$
0	1	0	$P_2 = \overline{A}B\overline{C}$	$S_2 = A + \overline{B} + C$
0	1	1	$P_3 = \overline{A}BC$	$S_3 = A + \overline{B} + \overline{C}$
1	0	0	$P_4 = A\overline{B}\,\overline{C}$	$S_4 = \overline{A} + B + C$
1	0	1	$P_5 = A\overline{B}C$	$S_5 = \overline{A} + B + \overline{C}$
1	1	0	$P_6 = AB\overline{C}$	$S_6 = \overline{A} + \overline{B} + C$
1	1	1	$P_7 = ABC$	$S_7 = \overline{A} + \overline{B} + \overline{C}$

Fig. 48. *Términos P y S de una función de tres variables A, B y C.*

Los índices de los términos P y S corresponden a los valores decimales de las combinaciones de las variables. De los dos cuadros precedentes, se sacan inmediatamente las siguientes relaciones:

$$\overline{A}\,\overline{B} + \overline{A}B + A\overline{B} + AB = 1\,,$$
$$(A + B)(A + \overline{B})(\overline{A} + B)(\overline{A} + \overline{B}) = 0\,,$$

y, para una función de tres variables:

$$\overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C} + A\overline{B}C + AB\overline{C} + ABC = 1\,,$$
$$(A + B + C)(A + B + \overline{C})(A + \overline{B} + \overline{C})(\overline{A} + B + C)$$
$$(\overline{A} + B + \overline{C})(\overline{A} + \overline{B} + C)(\overline{A} + \overline{B} + \overline{C}) = 0\,.$$

La suma de todas las combinaciones para las cuales la función 1 da evidentemente 1. Igualmente el producto de todas las combinaciones para las cuales la función es 0 es evidentemente 0. Lo que se expone mediante los siguientes teoremas:

Teorema 1

La suma lógica de los 2^n términos P de una función "booleana" de n variables es igual a 1:

$$\sum_{i=0}^{2^n-1} P_i = 1 \,. \tag{80}$$

Teorema 2

El producto lógico de los 2^n términos S de una función "booleana" de n variables es igual a 0:

$$\prod_{i=0}^{2^n-1} S_i = 0 \,. \tag{81}$$

Teorema 3

El producto lógico de dos términos P cualesquiera de una función "booleana" es nulo:

$$P_i P_j = 0 \quad \text{para} \quad i \neq j \,. \tag{82}$$

Teorema 4

La suma lógica de cualquiera de los dos términos S de una función "booleana" es siempre igual a 1:

$$S_i + S_j = 1 \quad \text{para} \quad i \neq j \,. \tag{83}$$

Estos dos últimos teoremas se demuestran fácilmente. En efecto, cada término P o S de una función "booleana" difiere al menos en una variable: el producto es nulo y la suma igual a 1.

Ejemplo. Volvamos a la tabla de la figura 48. Tenemos, por ejemplo:

$$P_1 P_5 = \bar{A}\bar{B}C \,.\, A\bar{B}\bar{C} = 0 \,,$$
$$S_1 + S_5 = (A + B + \bar{C}) + (\bar{A} + B + \bar{C}) = 1 \,.$$

Teorema 5

Una función "booleana" de n variables puede representarse por una y sólo una suma lógica de términos P:

$$F = \sum_{i=0}^{2^n-1} \alpha_i P_i \,. \tag{84}$$

Teorema 6

Una función "booleana" de n variables puede representarse por un único producto lógico de términos S:

$$F = \prod_{i=0}^{2^n-1} (\alpha_i + S_i) \tag{85}$$

α_i representa lo que se llama *los números característicos* de una función "booleana". Si estos números están divididos en partes iguales de ceros y unos, la función se llama *neutra.*

Ejemplo. Consideremos una función F_1 de las tres variables tales que

$$F_1 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + AB\bar{C} .$$

Viendo la tabla de la figura 48, podemos escribir

$$F_1 = P_1 + P_2 + P_6 ,$$

de donde se saca según (84):

$$\alpha_1 = \alpha_2 = \alpha_6 = 1 .$$

Como hay tres variables, *A, B* y *C,* la función completa F_1 tiene $2^3 = 8$ términos *P.* Los términos que faltan se caracterizan por

$$\alpha_0 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_7 = 0 .$$

La aplicación de la fórmula (85) da:

$$F_1 = S_0 \quad S_3 \quad S_4 \quad S_5 \quad S_7 .$$

Se obtiene así la función F_1 bajo la fórmula de un producto de términos *S.* Estos se dan en la tabla de la figura 48.

Teorema 7

El complemento (inversión) de una función "booleana" de n variables puede representarse por una única suma lógica de términos P:

$$\bar{F} = \sum_{i=0}^{2^n-1} \bar{\alpha}_i P_i . \tag{86}$$

Teorema 8

El complemento de una función "booleana" de n variables puede representarse por un único producto o lógica de términos S:

$$\bar{F} = \prod_{i=0}^{2^n-1} (\bar{\alpha}_i + S_i) . \tag{87}$$

Consideremos una función "booleana" de n variables representada por (85) de la cual se toma el complemento:

$$\overline{F} = \overline{\prod_{i=0}^{2^n-1} (\alpha_i + S_i)} = \sum_{i=0}^{2^n-1} \overline{\alpha}_i \overline{S}_i$$

y según (50):

$$\overline{F} = \sum_{i=0}^{2^n-1} \overline{\alpha}_i P_i \,. \tag{88}$$

De la misma forma si

$$F = \sum_{i=0}^{2^n-1} \alpha_i P_i \tag{89}$$

obtenemos:

$$\overline{F} = \overline{\sum_{i=0}^{2^n-1} \alpha_i P_i} = \prod_{i=0}^{2^n-1} \overline{\alpha_i P_i}$$

$$\overline{F} = \prod_{i=0}^{2^n-1} (\overline{\alpha}_i + \overline{P}_i)$$

y según (50):

$$\overline{F} = \prod_{i=0}^{2^n-1} (\overline{\alpha}_i + S_i) \,. \tag{90}$$

Ejemplo. Volvamos con la función anterior F_1:

$$F_1 = \overline{A}\overline{B}C + \overline{A}B\overline{C} + AB\overline{C} \,,$$

donde

$$\alpha_1 = \alpha_2 = \alpha_6 = 1 \,,$$
$$\alpha_0 = \alpha_3 = \alpha_4 = \alpha_5 = \alpha_7 = 0 \,.$$

Obtenemos el complemento como suma de términos P (88):

$$\overline{F}_1 = P_0 + P_3 + P_4 + P_5 + P_7$$

y bajo la forma de un producto de términos S:

$$F_1 = S_1 . S_2 . S_6 \,.$$

Teorema 9

La suma lógica de 2^n términos de una función "booleana" de n variables es igual a 1:

$$\sum_{i=0}^{2^n-1} P_i = 1 \,. \tag{91}$$

Teorema 10

El producto lógico de 2^n términos de una función "booleana" de n variables es igual a 0:

$$\prod_{i=0}^{2^n-1} S_i = 0 .$$

En efecto, podemos escribir:

$$1 = F + \overline{F} = \sum_{i=0}^{2^n-1} \alpha_i P_i + \sum_{i=0}^{2^n-1} \overline{\alpha}_i P_i = \sum_{i=0}^{2^n-1} P_i \qquad (92)$$

pues $\alpha_i + \overline{\alpha}_i = 1$. Podemos escribir igualmente:

$$0 = F\overline{F} = \prod_{i=0}^{2^n-1} (\alpha_i + S_i) \prod_{i=0}^{2^n-1} (\overline{\alpha}_i + S_i) = \prod_{i=0}^{2^n-1} S_i ,$$

pues $\alpha_i \, \overline{\alpha}_i = 0$.

6.2 Suma standard en forma binaria

Es frecuente representar todos los términos de una suma standard por sus equivalentes binarios. Así en el último ejemplo:

$$X = f(A, B, C, D) = \sum (1011, 0010, 1011, 1001) . \qquad (93)$$

Esta forma de representar una función X se utilizará en la simplificación de los circuitos por el método de Quine y de McCluskey.

7. Aplicación de la tabla de verdad: circuitos de codificación y decodificación

Para codificar o entender la transcripción, en forma binaria, de un número decimal. Lo que se efectúa mediante un código, es decir de una tabla o figura, por un lado la cifra que se va a codificar, por otro el equivalente binario que se le desea atribuir.

Existen varios tipos de código, algunos de ellos son muy utilizados en las calculadoras.

Citemos, entre las más corrientes: el código binario puro, el decimal codificado en binario (BCD); el código + 3, el código 2 entre 5, el código binario de Gray. Estos códigos se examinarán detalladamente en el capítulo 3.

La decodificación es la operación inversa de la anterior, es decir el pase de un número binario a decimal. Los circuitos de codificación se realizan mediante

circuitos O, los circuitos de decodificación con circuitos Y. Los siguientes ejemplos permitirán una mejor comprensión del procedimiento.

Ejemplo 1. *Conversión de un número decimal en número binario*

En este código de conversión se representa cada cifra del número decimal por su equivalente binario. Como cada cifra del número decimal puede estar representado entre 0 y 9 en forma binaria estará comprendida entre 0000 y 1001.

El código utilizado es el siguiente:

Decimal	Binario
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Fig. 49. *Código* BCD.

Para realizar este código hay que disponer de un circuito de diez entradas correspondientes a los diez valores decimales de 0 a 9 y cuatro salidas correspondientes a cada potencia del equivalente binario, es decir, $2^0, 2^1, 2^2, 2^3$. Cada potencia podrá tomar dos valores, 0 ó 1, según sea el número decimal a la entrada.

Tendremos entonces ocho salidas, dos para cada potencia. El circuito de codificación correspondiente es el de la figura 50. Se utilizará la salida 1 de cada circuito O.

Utilizando la salida cero, se obtendrá el complemento restringido del número (ver Cap. 3).

Para decodificar un número decimal codificado en binario, se utilizará el circuito de la figura 51. Se dispondrá a la entrada de dos líneas para cada potencia del número binario, en total ocho líneas de entrada de dos líneas para cada potencia del número binario, en total ocho líneas de entrada. Evidentemente tendremos a la salida diez líneas correspondientes a las diez cifras de 0 a 9. De forma general, en un decodificador habrá dos veces más hilos de entrada que el número binario máximo a decodificar: un hilo para el valor verdadero, otro para el complementario.

Así, por ejemplo, para contar de 0 a 7 hay tres pares de salida: una salida verdadera y la complementaria. Disponemos de seis hilos de entrada. En la salida, tendremos ocho hilos correspondientes a los números del 0 al 7 que se desean decodificar.

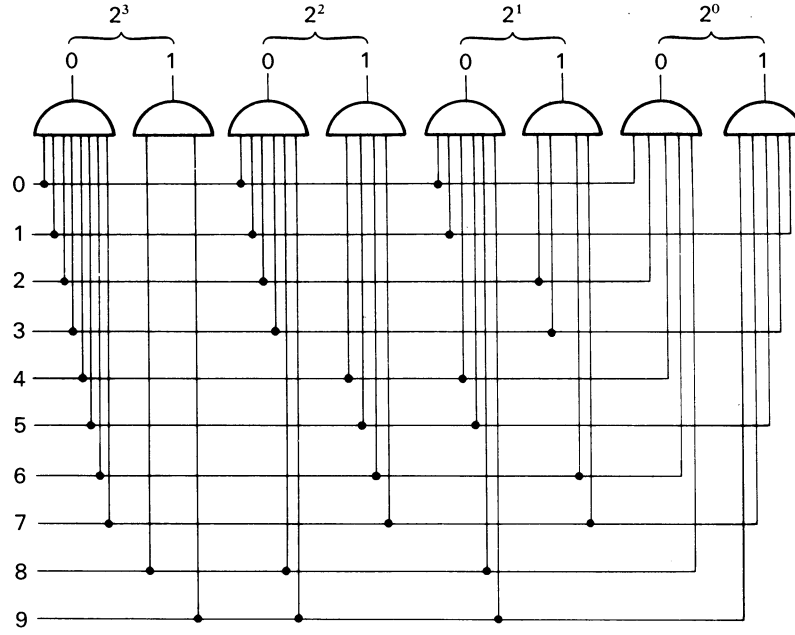


Fig. 50. *Circuito de codificación decimal → decimal codificado en binario.*

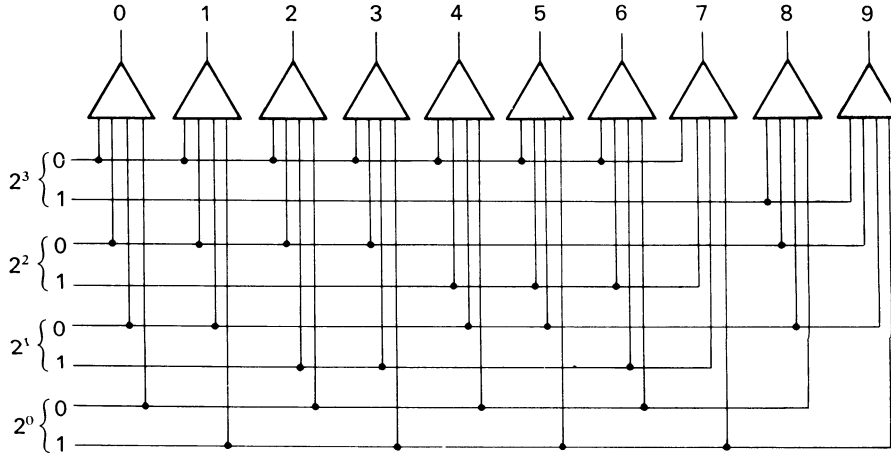


Fig. 51. *Decodificador decimal codificado en binario → decimal.*

Ejemplo 2. *Conversión del código Gray al código binario.*

El código Gray o código binario reflejado se utilizará particularmente en las entradas-salidas del ordenador. Una sola potencia cambia de estado al pasar de un valor codificado al valor superior. Volveremos sobre este código en el capítulo aritmético binario. La siguiente tabla (Fig. 42) da la equivalencia entre el código Gray y el código decimal codificado en binario o código 1, 2, 4, 8.

Tabla de correspondencia código Gray → código BCD.

Décimal	Código Gray	BCD
	A B C D	*W X Y Z*
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 1	0 0 1 0
3	0 0 1 0	0 0 1 1
4	0 1 1 0	0 1 0 0
5	0 1 1 1	0 1 0 1
6	0 1 0 1	0 1 1 0
7	0 1 0 0	0 1 1 1
8	1 1 0 0	1 0 0 0
9	1 1 0 1	1 0 0 1
10	1 1 1 1	1 0 1 0
11	1 1 1 0	1 0 1 1
12	1 0 1 0	1 1 0 0
13	1 0 1 1	1 1 0 1
14	1 0 0 1	1 1 1 0
15	1 0 0 0	1 1 1 1

Fig. 52. *Equivalencia entre el código Gray y el código BCD.*

Para determinar el circuito de codificación escribimos sucesivamente cuales son las condiciones para que W, X, Y y Z sean 1.

La tabla de verdad da, con la simplificación haciendo uso de la relación

$$\overline{A}\,\overline{B} + AB = \overline{\overline{A}B + A\overline{B}} = \overline{A \oplus B}\,,$$

las relaciones siguientes:

$$W = A,$$
$$X = \overline{A}B + A\overline{B} = A \oplus B,$$
$$Y = \overline{C}X + C\overline{X} = C \oplus X,$$
$$Z = \overline{Y}D + Y\overline{D} = D \oplus Y.$$

La codificación se efectuará según la figura 53:

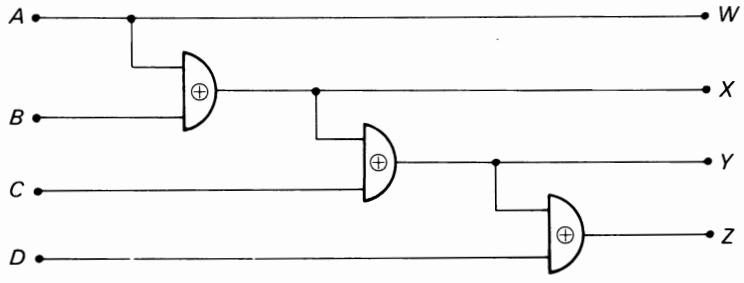


Fig. 53. *Paso del código Gray al código BCD.*

8. Simplificación de los circuitos

Una de las principales utilizaciones del álgebra de Boole es la simplificación de los circuitos. Esta simplificación puede efectuarse en diferentes momentos: en el momento del establecimiento del circuito o a posteriori cuando se posee un esquema del cual se ignora el origen y se desea simplificar. Es este último caso el que vamos a examinar. Otros métodos permiten igualmente simplificar los cálculos, en particular el método de los diagramas de Karnaugh y de Quine McCluskey. Estos métodos se estudiarán en el próximo capítulo.

Simplificar un circuito lógico es reducir el número de circuitos fundamentales O, Y e inversores que componen el circuito. Para simplificar este circuito se escribirá primeramente su función lógica examinando la composición del circuito. Una vez en posesión de su función lógica, es ella la que se simplifica mediante los métodos de álgebra de Boole. A partir de la función lógica simplificada se establecerá un nuevo esquema equivalente.

Ejemplo 1. *Simplificar el circuito de la figura* 54.

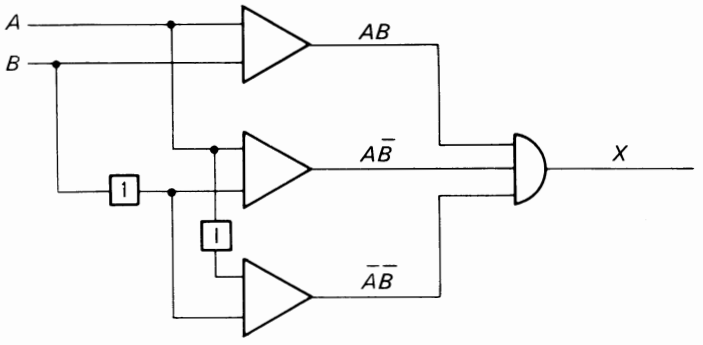


Fig. 54. *Circuito a simplificar.*

La función lógica realizada a la salida es:

$$X = AB + A\bar{B} + \bar{A}\bar{B}.$$

Esta función puede simplificarse así:

$$X = A(B + \bar{B}) + \bar{A}\bar{B},$$
$$X = A + \bar{A}\bar{B},$$
$$X = A + \bar{B},$$

donde se hace uso de las fórmulas fundamentales de la tabla de la figura 46. El circuito simplificado es un circuito O de dos entradas, A y $\bar{B}$ (Fig. 55).

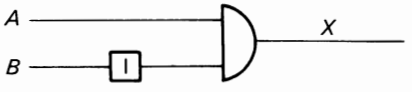


Fig. 55. *Esquema equivalente al de la figura 54.*

Ejemplo 2. *Simplificar el esquema equivalente de la figura 56.*

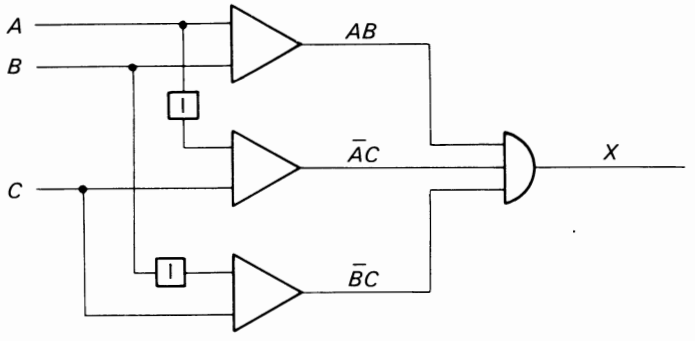


Fig. 56. *Circuito a simplificar.*

La función lógica realizada por el circuito es:

$$X = AB + \bar{A}C + \bar{B}C .$$

Saquemos C como factor común:

$$X = AB + C(\bar{A} + \bar{B}) ;$$

según el teorema de De Morgan:

$$X = AB + \overline{AB}.C ,$$

es decir, según este teorema $A + \bar{A}B = A + B$;

$$X = AB + C ;$$

que es el esquema de la figura 57:

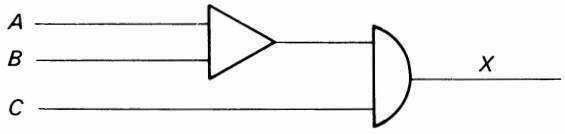


Fig. 57. *Circuito equivalente al de la figura 56.*

Ejemplo 3. *Simplificar el circuito de la figura 58. Los valores C y D están disponibles en sus valores complementarios.*

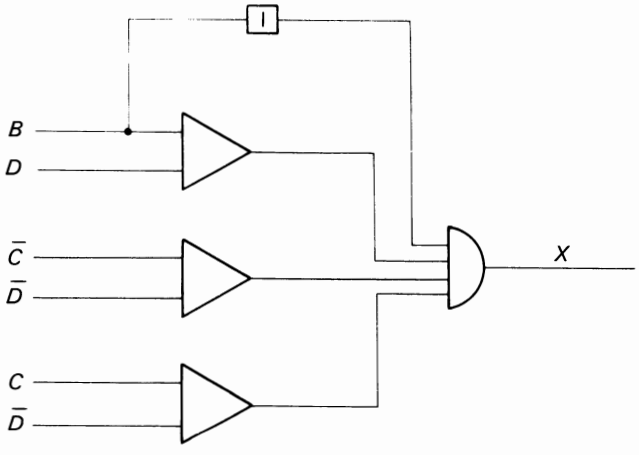


Fig. 58. *Circuito a simplificar.*

La función lógica realizada por el circuito es:

$$X = \bar{B} + BD + \bar{C}\bar{D} + C\bar{D} .$$

Podemos escribir, de acuerdo con las fórmulas fundamentales:

$$\bar{B} + BD = \bar{B} + D$$
$$\bar{D}(C + \bar{C}) = \bar{D},$$

de donde el valor de X:

$$X = \bar{B} + D + \bar{D},$$

por otra parte sabemos que:

$$D + \bar{D} = 1$$

de donde finalmente:

$$X = \bar{B} + 1 = 1$$

y el circuito equivalente es un cortocircuito (Fig. 59).

Fig. 59. *Circuito equivalente al de la figura 58.*

8.1. Caso en que se dispone un inversor a la salida

En este caso se calcula la función lógica a la salida *antes de su inversión*, X, a continuación se simplifica X de acuerdo con los métodos habituales. Con la ayuda de las fórmulas fundamentales, después se calcula $\bar{X}$.

Ejemplo. *Se desea simplificar el circuito de la figura 60. Se supone que los valores complementarios están disponibles.*

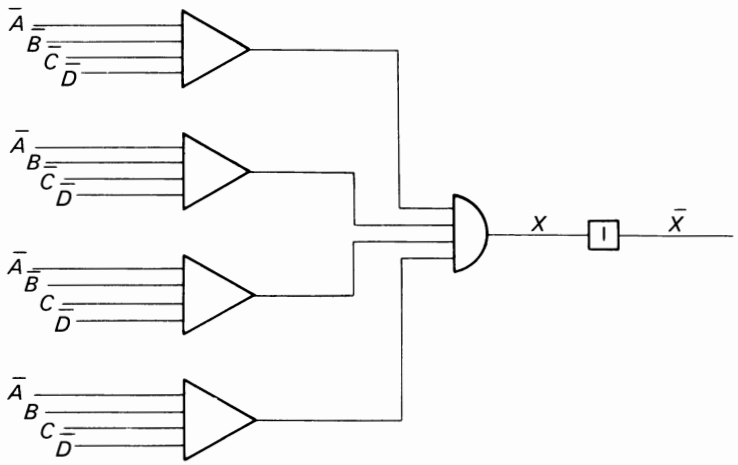


Fig. 60. *Circuito a simplificar.*

Se determinará no la función $\overline{X}$, sino la función lógica X antes de la inversión:

$$X = \overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + \overline{A}BC\overline{D}.$$

Simplifiquemos X, obtenemos:

$$X = \overline{A}\overline{C}\overline{D} + \overline{A}C\overline{D},$$
$$X = \overline{A}\overline{D}.$$

Invirtamos X para obtener $\overline{X}$:

$$X = A + D.$$

El circuito equivalente se reduce a un circuito O (Fig. 61):

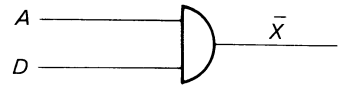


Fig. 61. *Esquema equivalente al de la figura 60.*

9. Los circuitos lógicos de memoria

En los circuitos estudiados hasta aquí, la aparición de los datos a la entrada de un circuito lógico se traduce a la salida por la aparición de una función lógica entre estos datos. Cuando desaparecen los datos a la entrada, desaparece la función lógica a la salida. En un ordenador se necesita siempre conservar en memoria durante un cierto tiempo los datos presentes a la entrada de un circuito. La realización de estos circuitos se apoya en la utilización de básculas electrónicas. Entendemos por báscula electrónica un dispositivo susceptible de cambiar de estado bajo control y conservar este estado indefinidamente. Así una báscula tomará el estado 1 y lo conservará indefinidamente hasta la aparición de una nueva orden, que hará pasar la báscula al estado 0. Se distinguen cuatro tipos de básculas de acuerdo con el número de sus entradas.

9.1 Báscula Trigger (báscula T)

Una báscula Trigger posee una entrada única T y dos salidas. Los tipos de básculas que estudiaremos, poseerán dos salidas: una salida verdadera dispuesta a la derecha, una salida complementaria dispuesta a la izquierda. El estado de la báscula lo da la observación de su salida verdadera. Si es 1 se dirá que está en estado 1, si es 0 se dirá que está en estado 0. Al mismo tiempo la salida comple-

mentaria proporcionará 0 si la báscula es 1 y 1 si la báscula es 0. El símbolo de una báscula T se da en la figura 62.

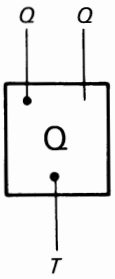


Fig. 62. *Representación simbólica de una báscula T.*

Esta báscula se llama Q y su salida verdadera Q.

Su salida complementaria dará $\overline{Q}$. La entrada de la báscula está referenciada por T. La báscula T realiza la función lógica siguiente: *cambia el estado cuando se aplica un 1 a su entrada cualquiera que sea su estado anterior y conserva este estado indefinidamente. No cambia de estado cuando se aplica* 0 *a su entrada T.* La tabla de verdad se da en la figura 63. Q_1 es el estado anterior antes de la aplicación de una variable "booleana" a su entrada T. *El nuevo estado de la báscula depende de su estado anterior.* La lectura de la tabla de verdad da para la salida verdadera:

$$Q_{t+1} = \overline{Q}_t\, T + Q_t\, \overline{T} = Q_t \oplus T \tag{94}$$

Q_t	T	Q_{t+1}
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 63. *Tabla de verdad de la báscula T.*

y como salida complementaria:

$$\overline{Q_{t+1}} = \overline{Q_t \oplus T} = Q_t \odot T\,. \tag{95}$$

Así una báscula T realiza un *O exclusivo* entre su estado anterior y la variable aplicada a su entrada T.

Para su salida falsa, produce la coincidencia en su estado anterior y la variable que se aplica a su entrada T.

9.2 Básculas J-K

En este tipo de básculas se dispone de dos entradas, J y K, y de las salidas como anteriormente, Q y $\overline{Q}$. La entrada J está destinada a la puesta a 1, la entrada K a la vuelta a 0, un 1 en su entrada J hará pasar a la báscula al estado 1. Si la báscula está en 1, un 1 a su entrada K hará pasar la báscula al estado 0 (puesta a cero: Raz).

Si la báscula está en 1, un 1 en su entrada J no cambiará el estado de la báscula, que conservará su estado anterior. Si la báscula está en el estado 0, un 1 en su entrada K no cambiará su estado que conservará el estado 0.

Finalmente, un 1 simultáneo en las dos entradas J y K cambiará el estado de la báscula cualquiera que sea su estado anterior. El símbolo de una báscula J-K se da en la figura 64 y su tabla de verdad, figura 65:

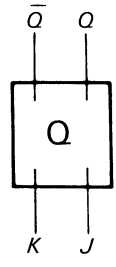


Fig. 64. *Símbolo de una báscula J-K.*

Q_t	J	K	Q_{t+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Fig. 65. *Tabla de verdad de la báscula J-K.*

La lectura de la tabla de verdad da para la salida verdadera:

$$Q_{t+1} = \overline{Q}_t J\overline{K} + \overline{Q}_t JK + Q_t \overline{J}\overline{K} + Q_t J\overline{K}$$

y para su salida complementaria:

$$\overline{Q_{t+1}} = \overline{Q}_t \overline{J}\overline{K} + \overline{Q}_t \overline{J}K + Q_t \overline{J}K + Q_t JK$$

después de la simplificación:

$$Q_{t+1} = J\overline{Q}_t + \overline{K}Q_t \tag{96}$$

$$\overline{Q_{t+1}} = \overline{J}\overline{Q}_t + KQ_t\,. \tag{97}$$

Estas dos últimas expresiones representan las funciones lógicas realizadas entre el estado anterior de la báscula y las variables J y K. La báscula conserva en memoria su nuevo estado Q_{t+1} hasta la aparición de los nuevos datos en sus entradas J y K.

9.3 Básculas R-S

Las básculas R-S son básculas J-K con una condición no permitida $RS = 0$. Las entradas se denominan R (Reset) y S (Set) que corresponden respectivamente a la situación 0 y 1 de la báscula. La aparición de dos 1 a las entradas R y S da una indeterminación: la báscula pasará indiferentemente al estado 0 ó al 1. La eliminación de esta condición equivale a escribir:

$$RS = 0 . \tag{98}$$

El esquema simbólico de la báscula R-S se da en la figura 66:

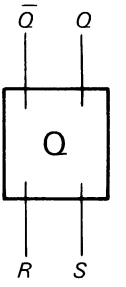


Fig. 66. *Símbolo de una báscula R-S.*

La tabla de verdad de la báscula R-S se da en la figura 67:

Q_t	R	S	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
1	0	0	1
1	0	1	1
1	1	0	0

Fig. 67. *Tabla de verdad de la báscula R-S.*

La lectura de esta tabla da, para la salida verdadera:

$$Q_{t+1} = \overline{Q}_t \overline{R}S + Q_t \overline{R}\overline{S} + Q_t \overline{R}S ,$$

y para la salida complementaria:

$$\overline{Q_{t+1}} = \overline{Q}_t \overline{R}\overline{S} + \overline{Q}_t R\overline{S} + Q_t R\overline{S} .$$

Incorporando la condición (98):

$$Q_{t+1} = \overline{R}S(Q_t + \overline{Q}_t) + Q_t \overline{R}\overline{S} + RS ,$$
$$Q_{t+1} = S + \overline{S}\overline{R}Q_t ,$$

sea finalmente:

$$Q_{t+1} = S + \overline{R}Q_t . \tag{99}$$

Para la salida complementaria, se obtiene:

$$\overline{Q_{t+1}} = R\overline{S} + \overline{Q}_t \overline{R}\overline{S} + RS$$
$$\overline{Q_{t+1}} = R + \overline{R}\overline{S}\overline{Q}_t$$

es decir:

$$\overline{Q_{t+1}} = R + \overline{S}\overline{Q}_t . \tag{100}$$

La ecuación (99) significa que la báscula tomará el estado 1 si $S = 1$ o si el estado anterior era ya 1 en ausencia de 1 en R. Las expresiones (99) y (100) representan las funciones lógicas realizadas por la báscula R-S entre su estado anterior Q y sus entradas R y S. Se recordará que se ha eliminado la condición $R = S = 1$, pues conduce a un estado indeterminado de la báscula.

9.4 Báscula R-S-T

Una báscula R-S-T es una báscula R-S a la que se ha añadido una entrada complementaria T. Cambia de estado cualquiera que sea su estado anterior cuando se aplica un 1 a su entrada T. Las entradas R y S son idénticas a las de la báscula R-S. La condición $R = S = T = 1$ no se podrá producir nunca pues conduce a un estado indeterminado de la báscula, lo que se traduce por:

$$RST = 0 \tag{101}$$

y

$$RS = ST = RT = 0 .$$

El símbolo de la báscula *R-S-T* se da en la figura 68:

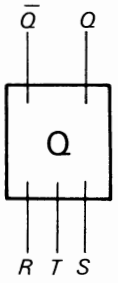


Fig. 68. *Símbolo de una báscula R-S-T.*

La tabla de verdad corresponde a la de la fig. 69.

Q_t	R	S	T	Q_{t+1}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	1	0	0	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	0

Fig. 69. *Tabla de verdad de la báscula R-S-T.*

La lectura de la tabla de verdad da, para la salida verdadera:

$$Q_{t+1} = \overline{Q}_t\,\overline{R}S\overline{T} + \overline{Q}_t\,\overline{R}S\overline{T} + Q_t\,\overline{R}\,\overline{S}\,\overline{T} + Q_t\,\overline{R}S\overline{T}$$

y para la salida complementaria:

$$\overline{Q_{t+1}} = \overline{Q}_t\,\overline{R}\,\overline{S}\,\overline{T} + \overline{Q}_t\,R\overline{S}\overline{T} + Q_t\,\overline{R}\,\overline{S}T + Q_t\,R\overline{S}\overline{T}\,.$$

Estas fórmulas pueden simplificarse:

$$Q_{t+1} = \overline{R}S\overline{T} + \overline{Q}_t\,\overline{R}\,\overline{S}\,\overline{T} + Q_t\,\overline{R}\,\overline{S}\,\overline{T}\,,$$

$$Q_{t+1} = \overline{R}\,\overline{T}(S + \overline{S}Q_t) + \overline{Q}_t\,\overline{R}\,\overline{S}T$$

$$= \overline{R}\,\overline{T}(S + Q_t) + \overline{Q}_t\,\overline{R}\,\overline{S}T = \overline{R}\,\overline{T}S + \overline{R}\,\overline{T}Q_t + \overline{Q}_t\,\overline{R}\,\overline{S}T\,.$$

Añadamos al miembro de la derecha $ST = 0$:

$$Q_{t+1} = \overline{R}\overline{T}Q_t + S\overline{R}\overline{T} + T(S + \overline{S}\overline{R}\overline{Q}_t) = \overline{R}\overline{T}Q_t + S\overline{R}\overline{T} + T(S + \overline{R}\overline{Q}_t),$$

$$Q_{t+1} = \overline{R}\overline{T}Q_t + S\overline{R}\overline{T} + ST + \overline{R}\overline{Q}_t T,$$

$$Q_{t+1} = \overline{R}\overline{T}Q_t + \overline{R}\overline{Q}_t T + S(T + \overline{T}\overline{R}),$$

$$Q_{t+1} = \overline{R}\overline{T}Q_t + \overline{R}\overline{Q}_t T + S(T + \overline{R}) = \overline{R}\overline{T}Q_t + \overline{R}\overline{Q}_t T + ST + S\overline{R}.$$

Añadamos al miembro de la derecha $RT = 0$:

$$Q_{t+1} = \overline{R}\overline{T}Q_t + ST + S\overline{R} + T(R + \overline{R}\overline{Q}_t),$$

$$Q_{t+1} = \overline{R}\overline{T}Q_t + ST + S\overline{R} + T(R + \overline{Q}_t),$$

$$Q_{t+1} = \overline{R}\overline{T}Q_t + ST + S\overline{R} + RT + T\overline{Q}_t.$$

Añadamos al miembro de la derecha $RS = 0$ y hagamos $ST = RT = 0$:

$$Q_{t+1} = \overline{R}\overline{T}Q_t + T\overline{Q}_t + S(R + \overline{R}).$$

Sea finalmente:

$$Q_{t+1} = \overline{R}\overline{T}Q_t + T\overline{Q}_t + S. \tag{103}$$

Se demostrará igualmente para la salida complementaria:

$$\overline{Q_{t+1}} = \overline{S}\overline{T}\overline{Q}_t + TQ_t + R. \tag{104}$$

La tabla de la figura 70 resume las principales funciones lógicas de los cuatro tipos de básculas.

Tipo	Salida verdadera Q_{t+1}	Salida complementaria $\overline{Q_{t+1}}$
T	$Q_t \oplus T$	$Q_t \odot T$
J-K	$J\overline{Q}_t + \overline{K}Q_t$	$\overline{J}\overline{Q}_t + KQ_t$
R-S	$S + \overline{R}Q_t$	$R + \overline{S}\overline{Q}_t$
R-S-T	$S + T\overline{Q}_t + \overline{R}\overline{T}Q_t$	$R + TQ_t + \overline{S}\overline{T}\overline{Q}_t$

Fig. 70. *Invenciones lógicas realizadas por los diferentes tipos de básculas.*

Capítulo 2

Los diagramas de Karnaugh

1. Introducción

El método del capítulo anterior se basa en la utilización del álgebra de Boole. Otro método consiste en trasladar las ecuaciones a simplificar a rejillas con dos, tres, cuatro, cinco y eventualmente seis variables. Se procede gráficamente para obtener la solución simplificada.

Este método lleva el nombre de método de Karnaugh y las rejillas utilizadas se llaman normalmente diagramas de Karnaugh. Una vez se ha introducido la expresión "booleana" en los diagramas de Karnaugh, se simplifica mediante métodos gráficos.

Así se llega más rápidamente al resultado. Este método, no obstante, está limitado a las funciones booleanas que no exceden de seis variables, lo que es generalmente suficiente para las necesidades de la práctica. Para funciones "booleanas" que posean más de seis variables, se prefiere recurrir al álgebra de Boole y al método de Quine McCluskey.

2. Definición del diagrama de Karnaugh. Introducción de una expresión «booleana» en un diagrama de Karnaugh

El diagrama de Karnaugh es una rejilla que tiene un número de "cuadros" vacíos igual al número de combinaciones de las variables de la función booleana que se propone simplificar. Así un diagrama de Karnaugh de una variable dará origen a un solo caso y este caso podrá tomar el valor 0 ó 1 (Fig. 1*a*).

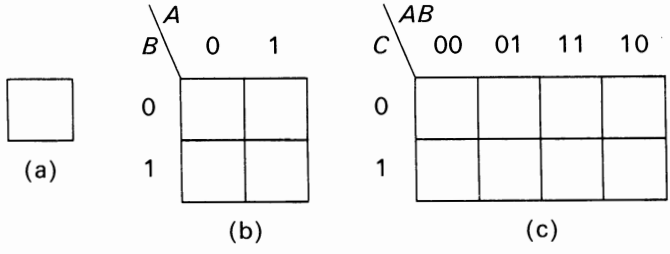
Un diagrama de Karnaugh con dos variables dará lugar a cuatro casos correspondientes a las cuatro combinaciones de las variables *A* y *B* (Fig. 1*b*),

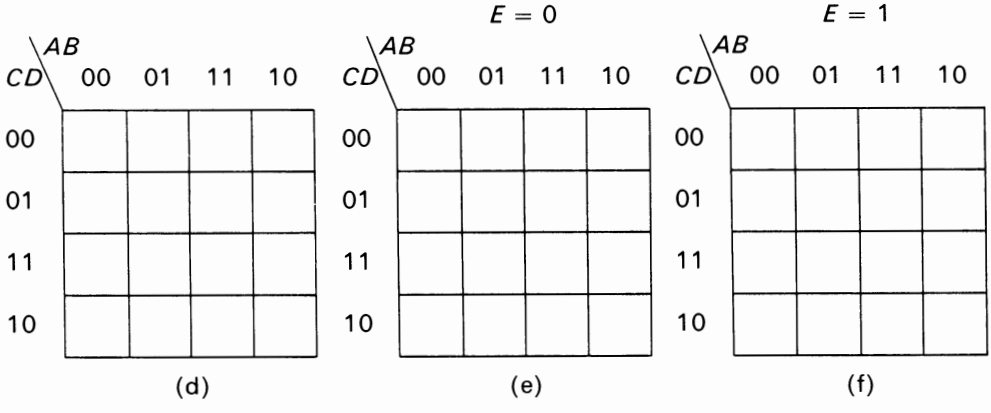
un diagrama de tres variables *A, B* y *C*, originará 8 casos,
un diagrama de cuatro variables *A, B, C* y *D*, originará 16 casos,
un diagrama de cinco variables *A, B, C, D* y *E*, originará 32 casos,
un diagrama de seis variables *A, B, C, D, E* y *F*, originará 64 casos.

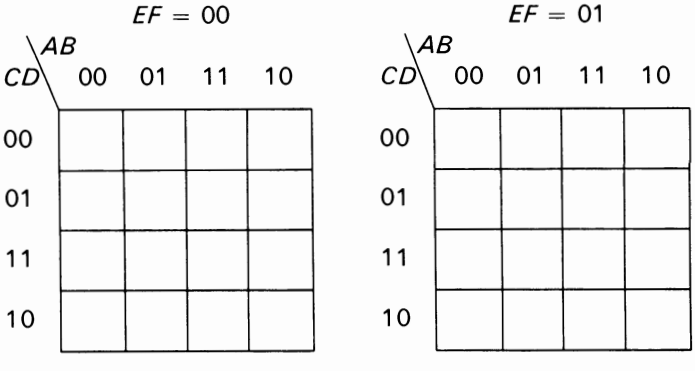
Los dos últimos casos se representarán en general, respectivamente, por dos y cuatro diagramas de dieciséis casos (Fig. 1*e* y *f*). Se lleva sobre los lados de cada diagrama las diferentes combinaciones de una o dos variables *A, AB, CD.*

En el caso de dos variables, el orden de sucesión de las diferentes combinaciones es 00, 01, 11, 10. *En esta sucesión una sola variable cambia de estado cuando se desplaza de un cuadro a otro.* Este orden deberá respetarse siempre si no se quiere tener graves errores.

Para introducir una función booleana en un diagrama de Karnaugh, se llevarán las combinaciones de las variables para las cuales la función es igual a 1, a los correspondientes cuadros del diagrama. Por convenio se dejan vacíos los cuadros del diagrama para los cuales la función tiene el valor 0. En algunos casos será útil el establecimiento de la tabla de verdad. En general se tratará de eliminar ésta y llevar directamente al diagrama de Karnaugh la función que se quiere simplificar.







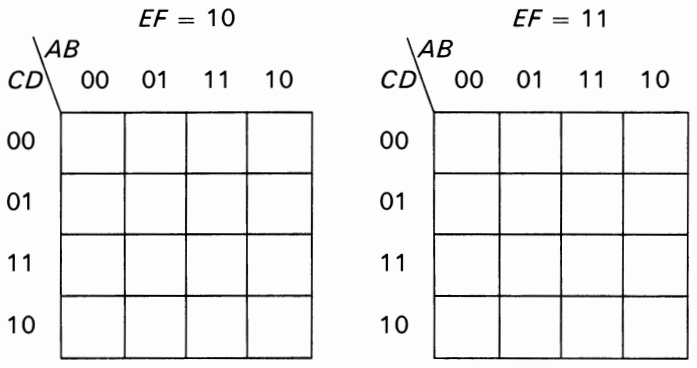


Fig. 1. *(a) Diagrama de Karnaugh de* 1 variable
(b) – – 2 variables
(c) – – 3 variables
(d) – – 4 variables
(e) – – 5 variables
(f) – – 6 variables.

Ejemplo 1. *Trasladar a un diagrama de Karnaugh, la función "booleana"* $X = AB$.

El diagrama de Karnaugh se limitará a dos variables. La función $X = 1$ si A y B son simultáneamente iguales a 1. Se introduce un 1 en el caso correspondiente a $A = 1$ y $B = 1$ (Fig. 2).

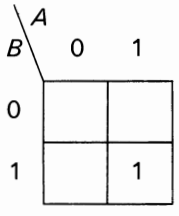


Fig. 2. $X = AB$.

Ejemplo 2. *Trasladar a un diagrama de Karnaugh la función "booleana"* $X = A + B$. *La tabla de verdad es la de la figura 3.*

A	B	$X = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 3. *Tabla de verdad de* $X = A + B$.

Hay tres condiciones para las cuales $X = 1$. Leyendo la tabla de verdad obtenemos:

$$X = \bar{A}B + A\bar{B} + AB$$

cuya forma simplificada es $X = A + B$. El diagrama de Karnaugh es el de la figura 4:

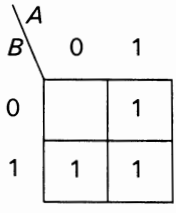


Fig. 4. $X = \bar{A}B + A\bar{B} + AB$.

Ejemplo 3. *Consideremos una función "booleana" de tres variables* $X = f(A, B, C)$ *tal que* $X = 1$ *si*

$$\begin{aligned} A &= 0, \quad B = 0, \quad C = 1 \\ A &= 0, \quad B = 1, \quad C = 0 \\ A &= 0, \quad B = 1, \quad C = 1 \\ A &= 1, \quad B = 0, \quad C = 1 \end{aligned}$$

$X = 0$ *para las restantes combinaciones de las variables. Trasladar la función* X *a un diagrama de Karnaugh de tres variables.*

Las condiciones enunciadas anteriormente se expresan mediante:

$$X = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C\,.$$

El diagrama de Karnaugh será de tres variables (Fig. 5). De los ocho casos cuatro serán 1:

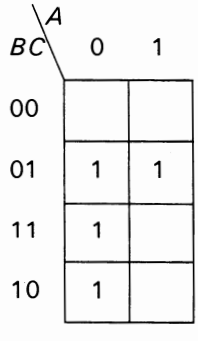


Fig. 5. $X = \bar{A}\bar{B}C + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C$.

Ejemplo 4. *Representar la función* $X = \bar{A} + A\bar{B}\bar{C} + BC$ *sobre un diagrama de Karnaugh.*

La función que se va a estudiar no está en la forma de suma standard. Dos términos están incompletos. En el primero faltan las variables B y C y en el último la variable A.

Podremos expresar X bajo la forma de una suma standard, multiplicando $\bar{A}$ por $(B + \bar{B})$ $(C + \bar{C}) = 1$ y BC por $A + \bar{A} = 1$. Normalmente se prefiere trasladar X directamente al diagrama de Karnaugh. Se procede entonces de la siguiente forma. El primer término A significa que $X = 1$ para $A = 0$ cualesquiera que sean los valores de B y C. Tendremos 1 en todos los casos correspondientes a la columna $A = 0$ pues $X = 1$ para todas las combinaciones de BC (00, 01, 11, 10). Igualmente el último término BC significa que $X = 1$ para $B = 1$ y $C = 1$. Cualesquiera que sean los valores de A. Tendremos un 1 en todos los casos que corresponden a la línea $BC = 11$. Recorriendo esta línea A varía de 0 a 1 (Fig. 6). Se introduce finalmente el último término $A\bar{B}\bar{C}$.

BC \ A	0	1
00	1	1
01	1	
11	1	1
10	1	

Fig. 6. $X = \bar{A} + A\bar{B}\bar{C} + BC$.

Ejemplo 5. *Sea una función "booleana" de cuatro variables* $X = f(A, B, C, D)$ *tal que* $X = 1$, *si:*

$$\begin{array}{llll} A = 0, & B = 0, & C = 1, & D = 0 \\ A = 0, & B = 1, & C = 0, & D = 0 \\ A = 1, & B = 1, & C = 1, & D = 0 \\ A = 1, & B = 0, & C = 0, & D = 1. \end{array}$$

La función es nula para las restantes combinaciones de las variables. Se pide representar la función en un diagrama de Karnaugh.

La función X se expresa mediante:

$$X = \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + ABC\bar{D} + A\bar{B}\bar{C}D,$$

que no expresa más que las condiciones comunicadas anteriormente. La expresión de X está bajo la forma de una suma standard. Se llevará sucesivamente cada uno de los términos al diagrama de Karnaugh (Fig. 7).

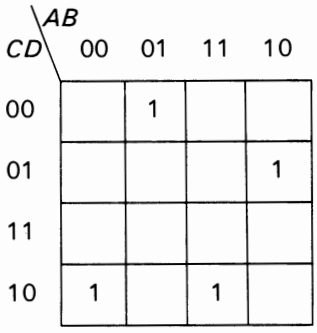


Fig. 7. $X = \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + ABC\bar{D} + A\bar{B}\bar{C}D$.

Ejemplo 6. *Llevemos a un diagrama de Karnaugh la siguiente expresión:*

$$X = \bar{A}B\bar{C}D + B\bar{C} + \bar{A}D\,.$$

El primer término está completo y corresponde a $A = 0$, $B = 1$, $C = 0$ y $D = 1$. El segundo término significa que $X = 1$ si $B = 1$ y $C = 0$ cualesquiera que sean los valores de A y D. El segundo término se situará en las dos columnas centrales del diagrama de Karnaugh donde $B = 1$ cualquiera que sea A, en la intersección de las líneas donde $C = 0$ cualquiera que sea D, es decir la primera y segunda línea.

Igualmente, el último término se sitúa en las columnas correspondientes a $A = 0$ cualquiera que sea B (primera y segunda columnas) en la intersección de las líneas donde $D = 1$ cualquiera que sea C (segunda y tercera líneas). La expresión $B\bar{C}$ introduce un nuevo 1 en el cuadro del primer término $\bar{A}B\bar{C}D$.

De cualquier manera sólo se escribirá un 1 en este caso, pues $1 + 1 = 1$. El diagrama de Karnaugh es el de la figura 8:

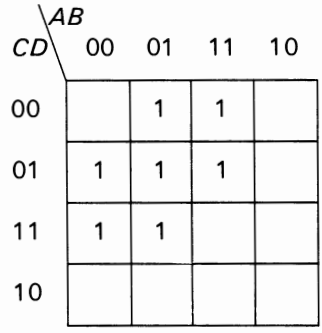


Fig. 8. $X = \bar{A}B\bar{C}D + B\bar{C} + \bar{A}D$.

Ejemplo 7. *Llevar a un diagrama de Karnaugh la función de cuatro variables* $X = AB + CD$.

La función $X = 1$ si $A = 1$ y $B = 1$ cualesquiera que sean los valores de C y D, o $C = 1$ y $D = 1$ cualesquiera que sean los valores de A y B. Tendremos 1 en todos los cuadros de la columna $AB = 11$ y en todos los cuadros de la línea $CD = 11$ (Fig. 9):

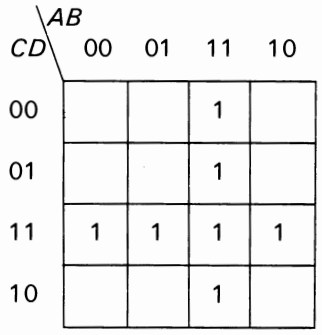


Fig. 9. $X = AB + CD$.

Ejemplo 8. *Llevar a un diagrama de Karnaugh la función de cuatro variables* $X = D$.

La función X posee 4 variables "booleanas" $X = f(A, B, C, D)$. Será 1 si $D=0$ cualesquiera que sean los valores de C y de AB. Se introduce un 1 en la primera y la última línea de diagrama de Karnaugh (Fig. 10).

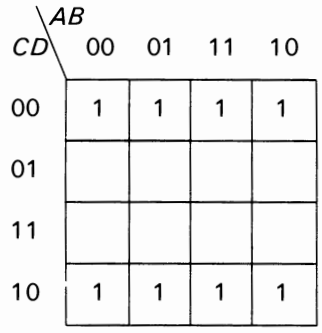


Fig. 10. $X = \overline{D}$.

Ejemplo 9. *Llevar a un diagrama de Karnaugh la función de cuatro variables* $X = \overline{A}$.

La función X será igual a 1 cuando $A = 0$ cualesquiera que sean los valores de B y CD. El diagrama de Karnaugh se indica en la figura 11:

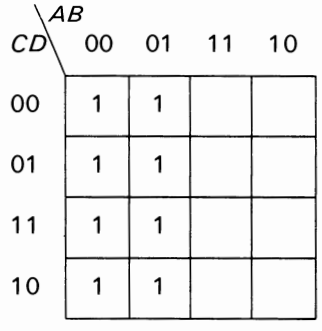


Fig. 11. $X = \overline{A}$.

Ejemplo 10. *Llevar a un diagrama de Karnaugh la función de cuatro variables* $X = \bar{B}\bar{D}$.

La función X será igual a 1 si $B = 0$ y $D = 0$ cualesquiera que sean los valores de C y A. Los valores $B = 0$ y $D = 0$ corresponden a los cuatro casos indicados con 1 en el diagrama de la figura 12. Se observará de nuevo que estos casos son independientes de los valores de A y C.

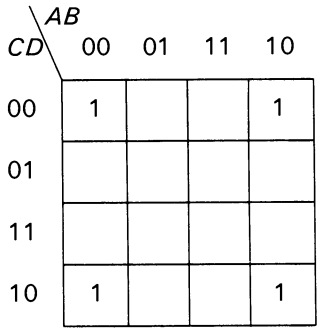


Fig. 12. $X = \bar{B}\bar{D}$.

Ejemplo 11. *Llevar a un diagrama de Karnaugh la función de cuatro variables* $X = BD$.

X será igual a 1 cuando $B = 1$ cualquiera que sea A, y $D = 1$ cualquiera que sea C. El diagrama de Karnaugh es entonces el de la figura 13.

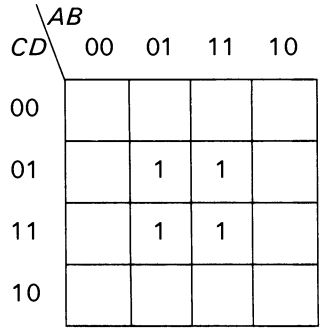


Fig. 13. $X = BD$.

Ejemplo 12. *Llevar a un diagrama de Karnaugh la función de cinco variables "booleanas":*

$$X = ABCDE + \bar{A}\bar{D} + \bar{A}B\bar{C}D\bar{E} + BCD + \bar{A}\bar{B}\bar{C}\bar{D}E.$$

En esta función de cinco variables tendremos dos diagramas de Karnaugh de cuatro variables: uno para el valor $E = 0$, y otro para el valor $E = 1$. Procediendo como en los casos anteriores obtenemos el diagrama de la figura 14.

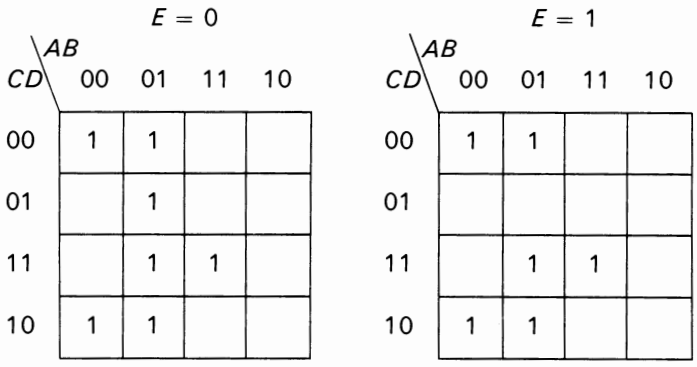


Fig. 14. $X = ABCDE + \bar{A}\bar{D} + \bar{A}B\bar{C}D\bar{E} + BCD + \bar{A}\bar{B}\bar{C}\bar{D}E$.

Ejemplo 13. *Llevar a un diagrama de Karnaugh la función de 5 variables* $X = \bar{E} + \bar{A}\bar{D}E$.

El primer término significa que $X = 5$ si $E = 0$ cualesquiera que sean los valores de las combinaciones de *A, B, C, D.* Se introducirá un 1 en todos los casos del diagrama correspondiente a $E = 0$. Para el segundo término se procederá como anteriormente. El diagrama de Karnaugh será finalmente el de la figura 15.

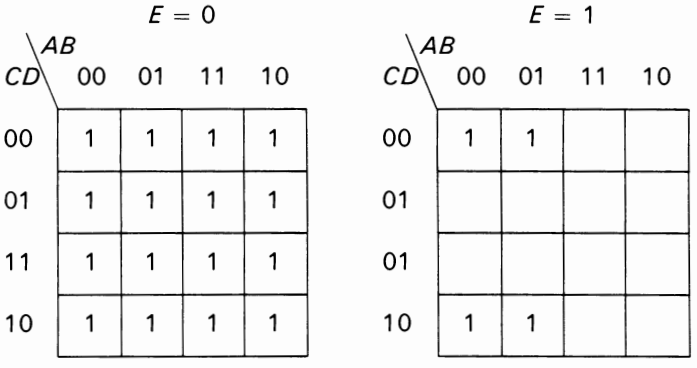


Fig. 15. $X = \bar{E} + \bar{A}\bar{D}E$.

Ejemplo 14. *Llevar a un diagrama de Karnaugh la función de seis variables booleanas:*

$$X = ABCDEF + \bar{A}C\bar{D}F + \bar{E}\bar{F} + CDE\bar{F}.$$

Tendremos cuatro diagramas de Karnaugh asociados a cada combinación de las variables *EF*, así para el primer término $EF = 11$, para el segundo término $F = 1$ cualquiera que sea *E*, para el tercer término $EF = 00$ cualesquiera que sean los valores de las combinaciones de los valores *A, B, C, D.* Finalmente para el último término $EF = 10$. Cada una de estas combinaciones da origen a un diagrama de Karnaugh de cuatro variables. Para cada uno de estos diagramas se procederá como en los diagramas precedentes. El diagrama general es el de la figura 16.

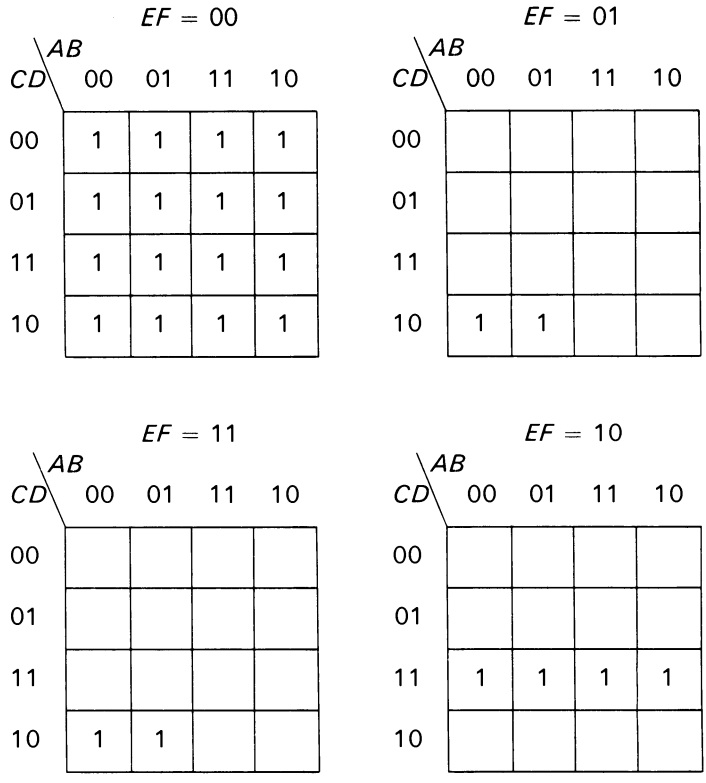


Fig. 16. $X = ABCDEF + \bar{A}C\bar{D}F + \bar{E}\bar{F} + CDE\bar{F}$.

2.1 Número de cuadrículas ocupadas en un diagrama de Karnaugh por un término de una función booleana

Consideremos una función de cuatro variables $X = f(A, B, C, D)$. Si uno de los términos de esta función comprende las cuatro variables bajo una forma directa o, inversa, le corresponde una sola cuadrícula del diagrama de Karnaugh. Veamos un ejemplo. Sea la función $X = ABCD$. Se representará en el diagrama de la figura 17 por una sola cuadrícula conteniendo un 1. Si la función contiene tres variables, por ejemplo $X = ABC$, dos cuadrículas del diagrama de Karnaugh contendrán un 1 (Fig. 18). Si la función sólo contiene dos variables, cuatro cuadrículas del diagrama se rellenarán con 1 (Fig. 19). Finalmente si la función contiene una sola variable $X = A$, ocho cuadrículas del diagrama representarán la función (Fig. 20).

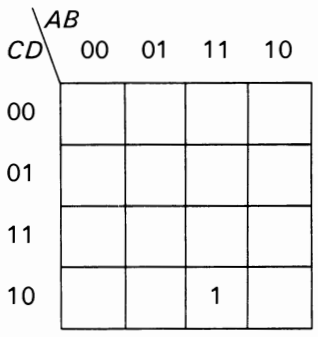


Fig. 17. $X = ABC\overline{D}$.

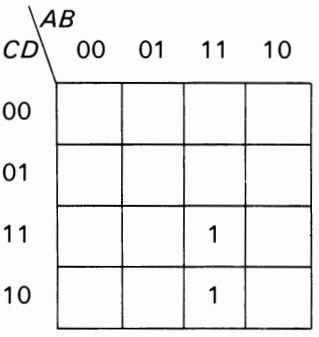


Fig. 18. $X = ABC$.

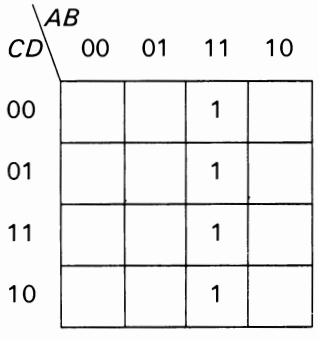


Fig. 19. $X = AB$.

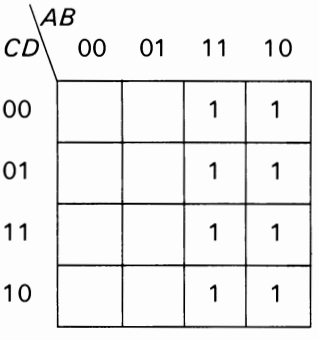


Fig. 20. $X = A$.

De una forma general el número de 1, es decir el número de cuadrículas del diagrama de Karnaugh necesarias para caracterizar un término de una función "booleana" se expresará mediante:

$$\text{Número de cuadrículas} = 2^N - Q$$

N : número de variables de la función,
Q : número de variables en el término que se desea introducir en el diagrama.

3. *Lectura del diagrama de Karnaugh. Simplificación de las funciones*

Una vez introducida la función booleana en un diagrama de Karnaugh se procederá a su lectura. Se simplificará la función agrupando los "1" contenidos en cuadrículas adyacentes. *Por cuadrículas adyacentes se entiende dos cuadrículas del*

diagrama de Karnaugh que sólo difieren en una variable. La figura 21 da un ejemplo de dos cuadrículas adyacentes. Su función representada es $X = ABCD + \overline{A}BCD$.

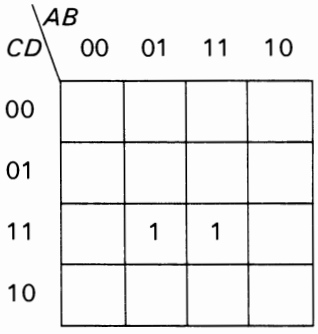


Fig. 21. $X = ABCD + \overline{A}BCD$.

Los dos términos de la función difieren únicamente en el valor de la variable A. Esta función se puede simplificar. En efecto, podemos escribir:

$$X = BCD(A + \overline{A}) = BCD \qquad (1)$$

recordando que $A + \overline{A} = 1$.

Cada vez que estemos en presencia de dos cuadrículas adyacentes, la función representada en el diagrama de Karnaugh podrá simplificarse mediante la aplicación, en el momento de la lectura, de la fórmula (1). La figura 22 da los diferentes casos de cuadrículas adyacentes. Según la fórmula (1), sólo se mantiene la lectura en las variables que conservan sus valores cuando se pasa de una cuadrícula a la otra. En el momento de la lectura se rodea a los "1" de las cuadrículas adyacentes mediante un bucle (Fig. 22), que indica que estos "1" se agrupan juntos para

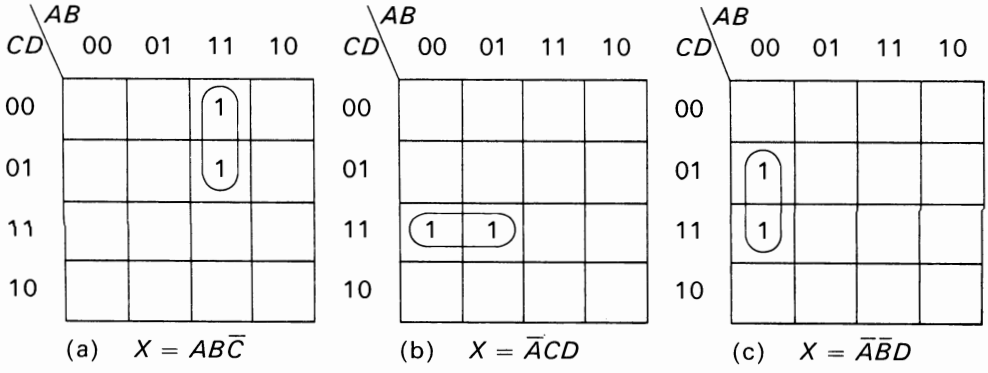


Fig. 22. *Diferentes casos de cuadrículas adyacentes.*

obtener una expresión simplificada de la función. Así el ejemplo *(a)* de la figura 22 da $X = AB\overline{C}$, el ejemplo *(b)* : $X = \overline{A}CD$, y el ejemplo *(c)* :

$$X = \overline{A}\,\overline{B}D\ .$$

En un mismo diagrama pueden existir varios "1" adyacentes. En este caso, se efectuarán varios bucles y se hará la suma de los resultados simplificados obtenidos (Fig. 23).

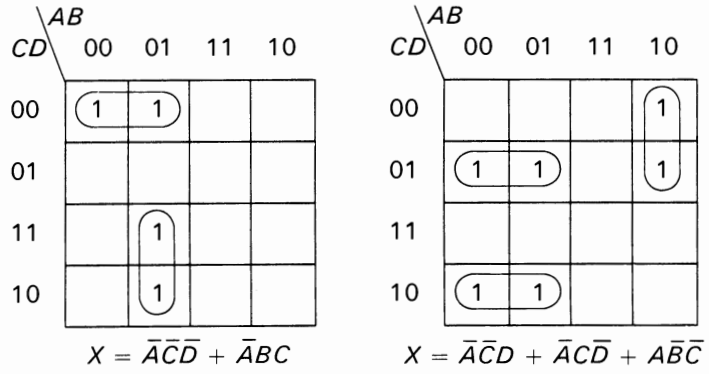


Fig. 23. *Diagramas de Karnaugh de varios bucles.*

3.1 Algunas cuadrículas adyacentes particulares

Dos cuadrículas adyacentes no están situadas obligatoriamente una junto a la otra. Viendo las figuras 24 y 25 se observará que las cuadrículas situadas próximas a los bordes exteriores del diagrama son igualmente adyacentes. Esto equivale a decir que se puede considerar al diagrama de Karnaugh como rodeando a un cilindro de diámetro tal que los bordes superior e inferior o derecha e izquierda se unen. En efecto, en la figura 24, la función representada es:

$$X = \overline{A}B\overline{C}\,\overline{D} + \overline{A}BC\overline{D}\ .$$

Los dos términos de la función sólo se diferencian en el valor de una variable, *C*. Después de establecer el bucle, la función tiene el valor simplificado

$$X = \overline{A}B\overline{D}(C + \overline{C}) = \overline{A}B\overline{D}\ .$$

Igualmente la función de la fig. 24*b* es $X = A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$ que da después de la asociación $X = A\bar{B}\bar{D}$.

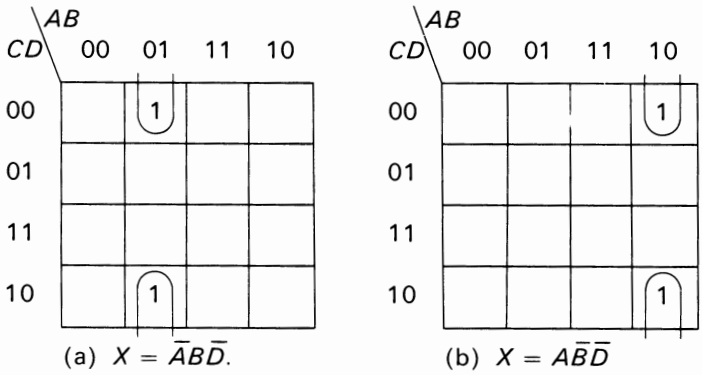


Fig. 24. *Otras cuadrículas adyacentes.*

Los bucles que rodean a los "1" están abiertos indicando que los extremos del bucle superior se unen a los extremos del bucle inferior para formar un único bucle.

La figura 25 da otros dos ejemplos de cuadrículas adyacentes que se obtienen *enrollando* el diagrama de Karnaugh alrededor de un eje vertical sobre un cilindro. En el ejemplo de la figura 24 el eje del cilindro era horizontal. En el caso de la figura 25 las cuadrículas extremas a la izquierda y a la derecha son adyacentes.

En el caso de la figura 25*a* la función representada en el diagrama es:

$$X = \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{C}D ,$$

en la que cada uno de los términos sólo difieren en la variable A. Se simplifica de la misma forma que anteriormente:

$$X = \bar{B}\bar{C}D(A + \bar{A}) = \bar{B}\bar{C}D .$$

Igualmente para la figura 25*b*, que representa la función:

$$X = \bar{A}\bar{B}C\bar{D} + A\bar{B}C\bar{D}$$

que una vez simplificada se reduce a:

$$X = \bar{B}C\bar{D}(A + \bar{A}) = \bar{B}C\bar{D} .$$

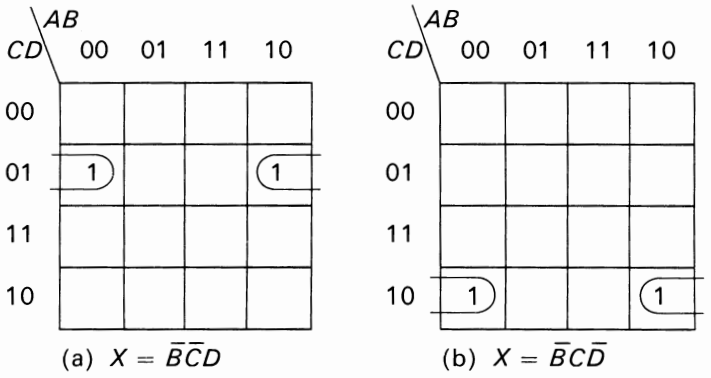


Fig. 25. *Otras cuadrículas adyacentes.*

3.2 Superposición de varios bucles

En algunos casos una misma cuadrícula puede englobarse varias veces en bucles diferentes. La figura 26 representa la función:

$$X = \bar{A}B\bar{C}D + AB\bar{C}D + ABCD \,. \tag{2}$$

Observaremos que el término $AB\bar{C}D$ sólo difiere en una variable con el primero *(A)* y el tercero *(C)*. De acuerdo con la regla $A + A = A$ no se cambia el valor de (2) añadiendo a X el término binario $ABCD$. Obtenemos en este caso:

$$X = \bar{A}B\bar{C}D + AB\bar{C}D + ABCD + AB\bar{C}D$$

y agrupando los términos dos a dos:

$$X = B\bar{C}D + ABD \,.$$

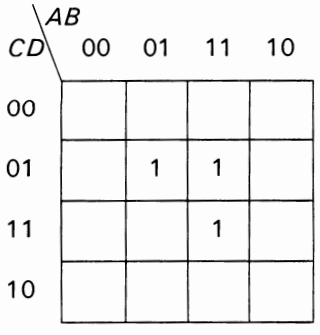


Fig. 26. $X = \bar{A}B\bar{C}D + AB\bar{C}D + ABCD$.

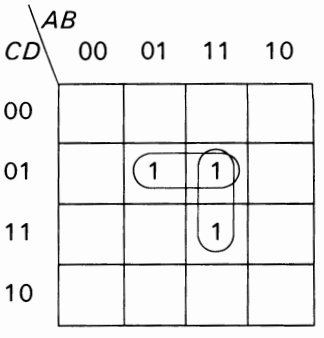


Fig. 27. $X = B\bar{C}D + ABD$.

La adición del término $AB\overline{C}D$ equivale a rodear dos veces la cuadrícula correspondiente del diagrama de Karnaugh en dos bucles diferentes. La figura 27 representa el diagrama de Karnaugh después de establecer los bucles.

La figura 28 da un ejemplo de una cuadrícula rodeada por cuatro bucles diferentes. Se señalará que la lectura *no simplificada* del diagrama de Karnaugh se traduce siempre por una suma standard.

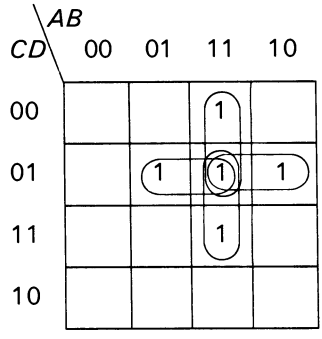


Fig. 28. $X = AB\overline{C} + ABD + B\overline{C}D + A\overline{C}D$.

3.3 Alargamiento de los bucles al mayor número de cuadrículas adyacentes

Hemos visto anteriormente que cada vez que se suprime una variable de una función "booleana" se multiplican por dos el número de "1" en el diagrama de Karnaugh. Estos "1" están dispuestos siempre en un grupo de cuadrículas adyacentes. Entonces es posible obtener la solución más simple, es decir que comprenda el mínimo de variables "booleanas" para la expresión de la función estudiada englobando en el bucle el máximo de cuadrículas adyacentes.

Consideremos el siguiente ejemplo:

$$X = \overline{A}B\overline{C}D + AB\overline{C}D + \overline{A}BCD + ABCD\ .$$

Esta función se simplifica mediante los teoremas del álgebra de Boole:

$$X = (A + \overline{A})(C + \overline{C})\,BD$$
$$X = BD\ .$$

La figura 29 da la representación de la función X en el diagrama de Karnaugh:

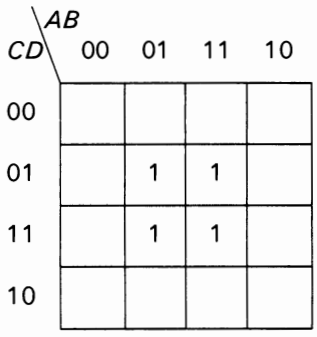


Fig. 29. $X = \bar{A}B\bar{C}D + AB\bar{C}D + \bar{A}BCD + ABCD.$

Se observará que una primera simplificación consiste en agrupar las cuadrículas adyacentes por parejas. Hay dos posibilidades agrupando horizontal o verticalmente las cuadrículas adyacentes, lo que corresponde a las figuras 30 y 31.

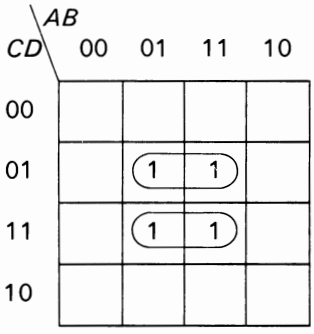


Fig. 30. $X = B\bar{C}D + BCD.$

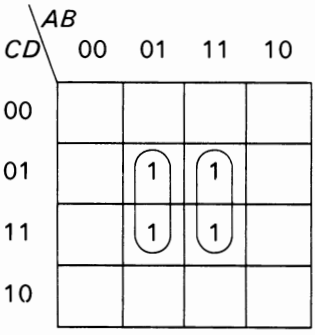


Fig. 31. $X = \bar{A}BD + ABD.$

Para cada una de estas figuras, la función simplificada tiene por valor:

$$X = B\bar{C}D + BCD,$$

$$X + \bar{A}BD + ABD.$$

Se observará que los dos términos de cada función difieren únicamente en el valor de una variable. Es entonces posible simplificar ventajosamente agrupando todas las cuadrículas adyacentes en un solo bucle (Fig. 32). Recorriendo el bucle, sólo se rodeará, como para los bucles más pequeños, las variables que conserven su estado. En estas condiciones se obtendrá $X = BD$.

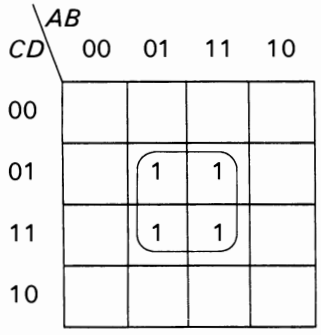


Fig. 32. $X = BD$.

Las figuras 33 a 38 dan ejemplos de reducción de funciones mediante bucles que rodean a cuatro cuadrículas adyacentes.

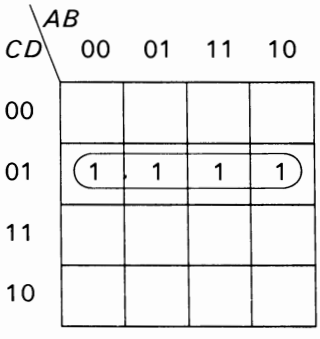


Fig. 33. $X = \overline{C}D$.

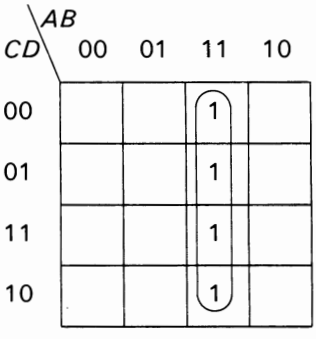


Fig. 34. $X = AB$.

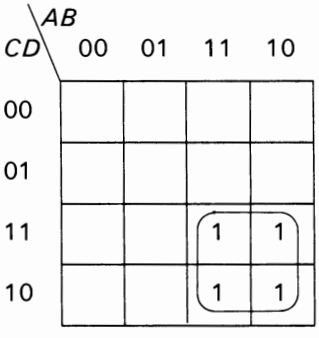


Fig. 35. $X = AC$.

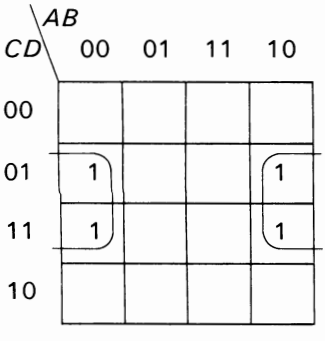


Fig. 36. $X = \overline{B}D$.

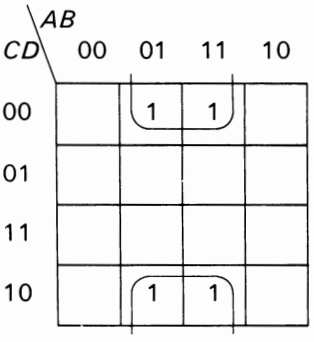


Fig. 37. $X = B\overline{D}$.

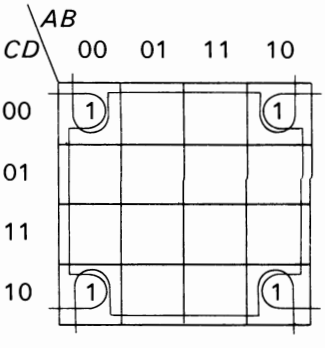


Fig. 38. $X = \overline{B}\overline{D}$.

Observaremos en la figura 38 la existencia de dos bucles diferentes que conducen al mismo resultado $X = BD$. Igualmente será posible rodear con un solo bucle ocho cuadrículas adyacentes del diagrama de Karnaugh. En este caso la función simplificada sólo comprenderá una variable. Las figuras 39 a 42 dan algunos ejemplos de bucles que rodean ocho cuadrículas adyacentes.

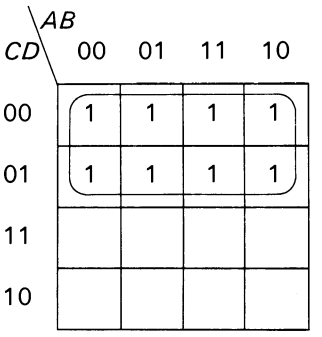


Fig. 39. $X = \overline{C}$.

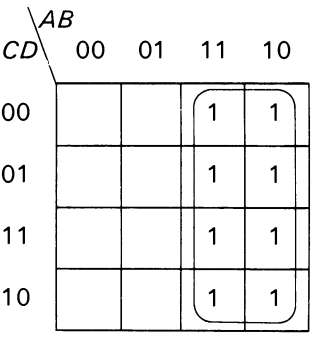


Fig. 40. $X = A$.

Las figuras 41 y 42 son dos ejemplos de cuadrículas adyacentes que se obtienen enrollando el diagrama de Karnaugh sobre un cilindro de eje horizontal (Fig. 41) y vertical (Fig. 42).

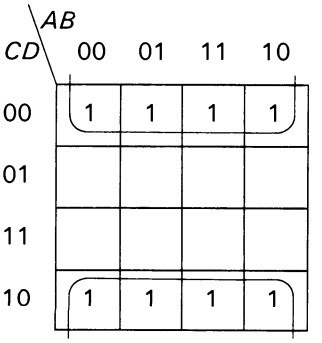


Fig. 41. $X = \overline{D}$.

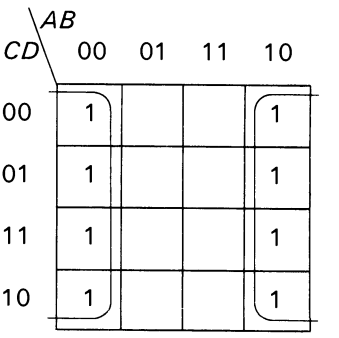


Fig. 42. $X = \overline{B}$.

Según la función, ésta puede simplificarse por varios bucles de magnitud diferentes. *En todos los casos se tratará de rodear con cada bucle el máximo de "1" contenidos en cuadrículas adyacentes del diagrama de Karnaugh. Dicho de otra forma, se intentará en todos los casos constituir bucles lo más grandes posibles.* Las figuras 43 a 46 dan varios ejemplos de bucles superpuestos. Se observará que

cada vez, se utilizan los mayores bucles posibles. Las figuras 47 y 48 dan un ejemplo de bucles inadecuados. En los dos casos se tiene el mismo número de bucles, tres en este caso.

Pero se observará que la solución de la figura 48 contiene un bucle más grande que da la solución más simple. En la figura 47 se obtiene.

$$X = A\overline{B}C\overline{D} + B\overline{C}D + A\overline{C}$$

y en la figura 48:

$$X = A\overline{B}\overline{D} + B\overline{C}D + A\overline{C}\,.$$

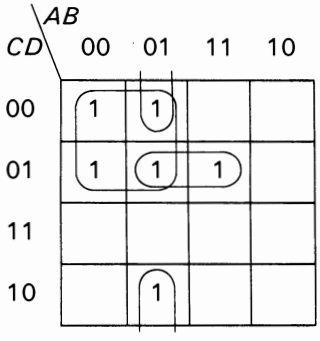


Fig. 43. $X=\overline{A}\overline{C}+B\overline{C}D+\overline{A}B\overline{D}$.

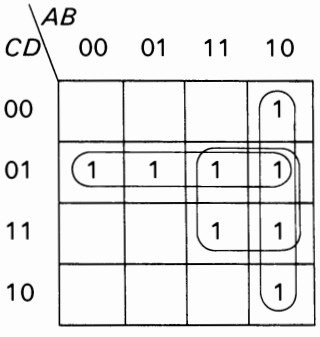


Fig. 44. $X=\overline{C}D+A\overline{B}+AD$.

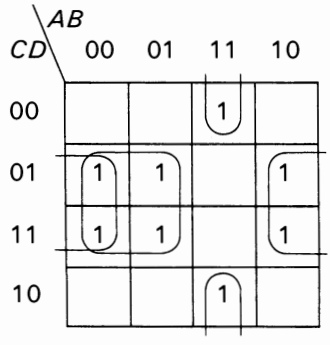


Fig. 45. $X=\overline{B}D+\overline{A}D+AB\overline{D}$.

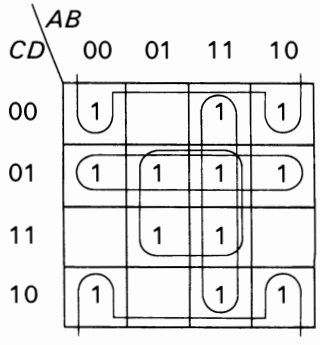


Fig. 46. $X=\overline{C}D+AB+BD+\overline{B}\overline{D}$.

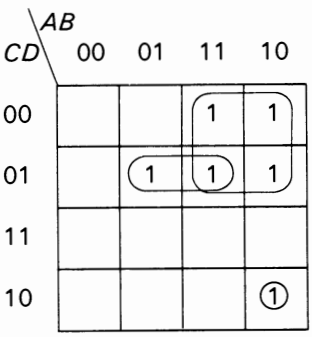


Fig. 47. $X=A\bar{B}C\bar{D}+B\bar{C}D+A\bar{C}$.

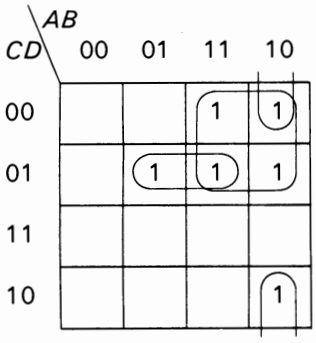


Fig. 48. $X=A\bar{B}\bar{D}+B\bar{C}D+A\bar{C}$.

3.4 Simplificación de los diagramas de Karnaugh de cinco y seis variables

La simplificación de los diagramas de Karnaugh de cinco y seis variables es más complicada. Consideremos inicialmente un diagrama de Karnaugh de cinco variables. Este se divide en dos diagramas de cuatro variables para los valores 0 y 1 de la variable E (Fig. 49). No se podrá, sin embargo, combinar entre ellas las cuadrículas adyacentes, salvo con la condición de que ocupen en los dos diagramas posiciones relativas idénticas.

Se pueden presentar varios casos:

- los dos diagramas de Karnaugh contienen bucles idénticos (Fig. 49);
- los dos diagramas contienen un bucle idéntico, integrado en uno de los diagramas, en un bucle más grande (Fig. 50);
- los dos diagramas contienen un bucle idéntico, incluido en un bucle más grande distinto para cada diagrama (Fig. 51).

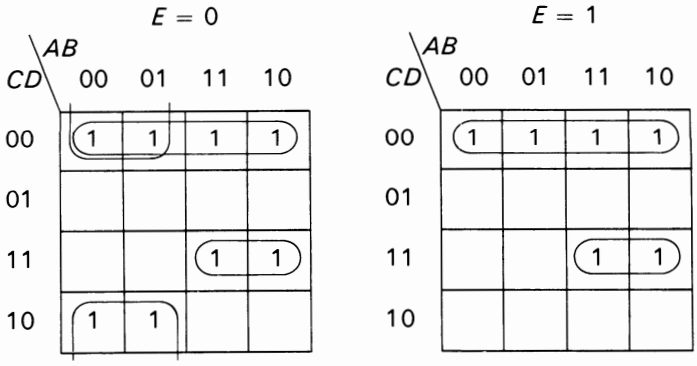


Fig. 49. $X = \bar{C}\bar{D} + ACD + \bar{A}\bar{D}\bar{E}$.

Así, en la fig. 49 los bucles son idénticos en los dos diagramas. Pueden combinarse entre ellos para eliminar la variable E. Un bucle en el diagrama $E = 0$ permanece; tendrá como valor $A\bar{D}\bar{E}$.

En la figura 50 un bucle es común a dos diagramas. Puede representarse mediante $A\bar{B}D$. Así se elimina la variable E. Los otros dos bucles no pueden combinarse y se leen directamente.

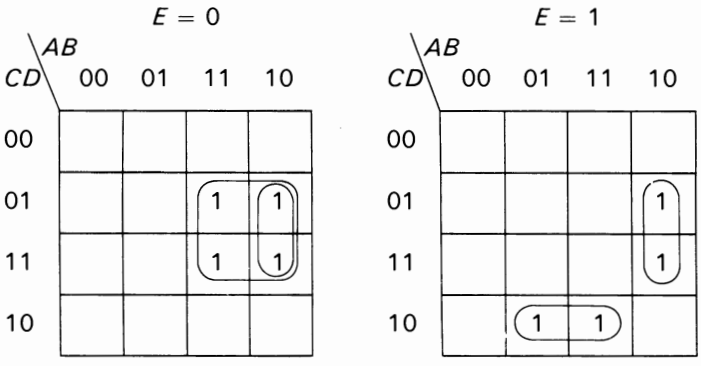


Fig. 50. $X = A\bar{B}D + \bar{E}AD + EBC\bar{D}$.

En la figura 51, dos bucles idénticos están incluidos en dos bucles distintos más grandes. Sólo se considerarán los bucles más grandes. Finalmente se añadirá el bucle que abarca a dos cuadrículas adyacentes en el diagrama relativo a $E = 0$.

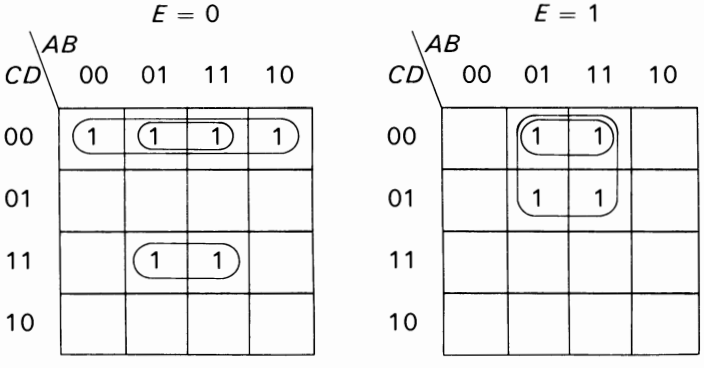


Fig. 51. $X = \bar{C}\bar{D}\bar{E} + B\bar{C}E + BCD\bar{E}$.

Consideremos ahora el caso de un diagrama de Karnaugh de seis variables A, B, C, D, E y F. Se compone, de acuerdo con lo que hemos visto, de cuatro sub-diagramas de cuatro variables $ABCD$ para cada una de las combinaciones de las dos variables $EF = 00, 01, 10, 11$.

Las combinaciones de cuadrículas adyacentes sólo podrán tener lugar entre dos diagramas horizontal o verticalmente. Para estas cuadrículas, es necesario que los valores de *EF* sólo difieran en una variable. Así, no es posible combinar las cuadrículas del diagrama correspondiente a $EF = 00$ con los dos diagramas $EF = 11$. Igualmente para los diagramas correspondientes a $EF = 10$ y $EF = 01$. En la figura 52 un bucle de dos cuadrículas es común a los cuatro diagramas que corresponden a $EF = 00$ y $EF = 10$: se combinarán juntas. Para los otros bucles se procederá como anteriormente.

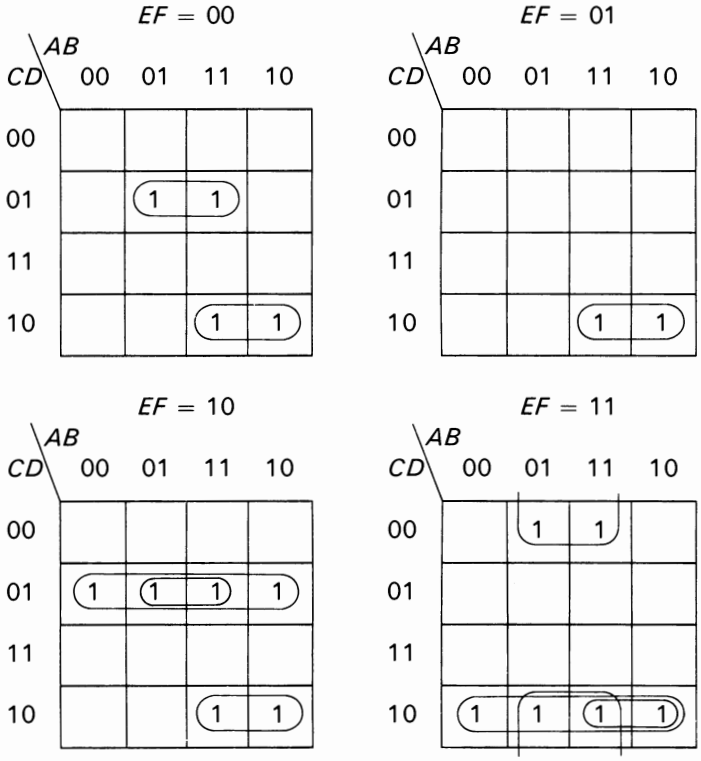


Fig. 52. $X = AC\bar{D} + B\bar{C}D\bar{F} + C\bar{D}EF + \bar{C}DE\bar{F} + B\bar{D}EF$.

3.5 Función "booleana" invertida en el diagrama de Karnaugh

Una aplicación importante del diagrama de Karnaugh es la determinación inmediata de la inversa de la función estudiada. Así si se ha llevado $X = f(A, B, C, D)$, sobre un diagrama de Karnaugh para obtener el diagrama de la función inversa $\bar{X}$, es suficiente sustituir los "1" por ceros y recíprocamente. Esto se deduce del hecho

de que $A + \bar{A} = 1$, $\bar{A} = 1 - A$. Así se ha llevado X al diagrama de Karnaugh, para obtener X es suficiente leer las cuadrículas vacías del diagrama según las reglas establecidas anteriormente. Recordemos que una cuadrícula vacía contiene un cero. La solución así obtenida para $\bar{X}$ será la solución más *simple.*

Ejemplo 1. *Sea una función "booleana" de cuatro variables* $X = f(A, B, C, D)$ *tal que* $X = 1$ *para las siguientes condiciones:*

$A = 0, \quad B = 0, \quad C = 0, \quad D = 0$
$A = 0, \quad B = 1, \quad C = 0, \quad D = 0$
$A = 1, \quad B = 1, \quad C = 0, \quad D = 0$
$A = 1, \quad B = 0, \quad C = 0, \quad D = 0$
$A = 0, \quad B = 1, \quad C = 0, \quad D = 1$
$A = 1, \quad B = 1, \quad C = 0, \quad D = 1$
$A = 0, \quad B = 0, \quad C = 1, \quad D = 0$
$A = 1, \quad B = 0, \quad C = 1, \quad D = 0.$

La función será nula para las restantes combinaciones de las variables. Se pide establecer la solución más simple para X y $\bar{X}$.

Llevemos la función X a un diagrama de Karnaugh de cuatro variables, figura 53.

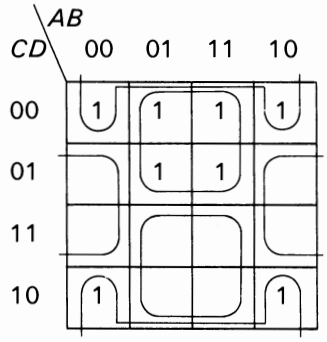


Fig. 53. $X = \bar{B}\bar{D} + B\bar{C}$
$\bar{X} = BC + \bar{B}D.$

La solución más simple sólo contiene dos bucles, lo que da la solución:

$$X = \bar{B}\bar{D} + B\bar{C}.$$

La solución más simple para $\bar{X}$ se obtendrá rodeando las cuadrículas adyacentes vacías. Esto se efectuará mediante dos bucles (Fig. 53), de donde se saca el valor de $\bar{X}$:

$$\bar{X} = BC + \bar{B}D$$

que representa la solución más simple para $\bar{X}$.

Ejemplo 2. *Simplificar la siguiente expresión "booleana", después determinar la expresión más simple de su inversa:*

$$X = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + AB\bar{C}\bar{D} + A\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}\bar{B}CD + \bar{A}BCD .$$

Llevemos esta expresión a un diagrama de Karnaugh y simplifiquemos (Fig. 54):

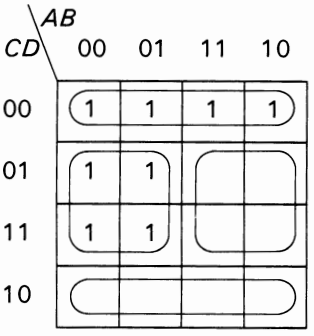


Fig. 54. $X = \bar{C}\bar{D} + \bar{A}D$
$\bar{X} = C\bar{D} + AD.$

Obtenemos:

$$X = \bar{C}\bar{D} + \bar{A}D .$$

Para determinar $\bar{X}$ se rodearán las cuadrículas adyacentes vacías del diagrama de Karnaugh:

$$\bar{X} = C\bar{D} + AD .$$

3.6 Aplicación de los diagramas de Karnaugh a la simplificación de los circuitos

La utilización de los diagramas de Karnaugh representa un método elegante y eficaz para la simplificación de los circuitos. Se procederá de la siguiente forma: a partir de un circuito lógico determinado escribir la ecuación "booleana" de la salida expresada en funciones variables de entrada. Simplificar a continuación trasladando el diagrama de Karnaugh mediante el método de los bucles y establecer el nuevo circuito.

Ejemplo 1. *Hallar un circuito equivalente simplificado del circuito siguiente* (Fig. 55):

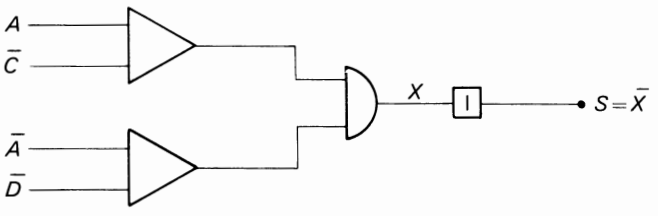


Fig. 55.

Tenemos, si X es la salida del circuito O:

$$S = \overline{X}.$$

Calculemos X y llevemos su valor al diagrama de Karnaugh (Fig. 56):

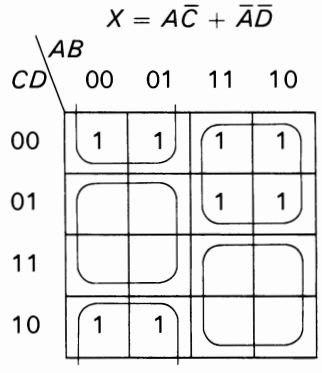


Fig. 56. $X = A\overline{C} + \overline{A}\overline{D}$
$\overline{X} = \overline{A}D + AC.$

Rodeando las cuadrículas vacías obtenemos:

$$S = \overline{X} = \overline{A}D + AC,$$

de donde el esquema de la figura 57 es equivalente al de la figura 55:

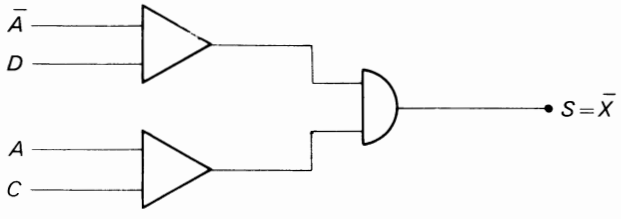


Fig. 57. *Esquema equivalente al de la figura 55.*

Ejemplo 2. *Simplificar el siguiente circuito* (Fig. 58)

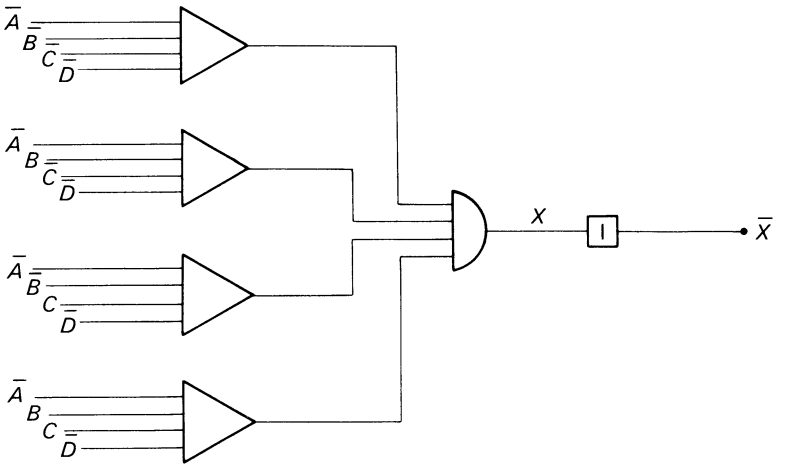


Fig. 58.

Calculemos primeramente X, después $\bar{X}$, tenemos:

$$X = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D}\,.$$

Llevemos X al diagrama de Karnaugh correspondiente (Fig. 59). Después de la simplificación obtenemos:

$$X = \bar{A}\bar{D}$$

y para $\bar{X}$ rodeando los ceros del diagrama:

$$S = \bar{X} = A + D\,.$$

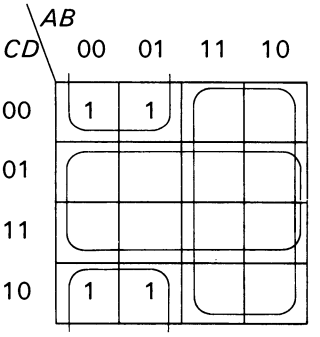


Fig. 59. $X = \bar{A}\bar{D}$ $\bar{X} = A + D$.

De donde el esquema equivalente de la figura 60, idéntico desde el punto de vista lógico al de la figura 58:

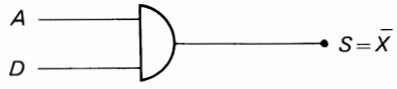


Fig. 60. *Esquema equivalente al de la figura 58.*

Se observará que el esquema de la figura 60 es mucho más económico que el de la figura 58.

Ejemplo 3. *Simplificar el siguiente circuito* (Figura 61):

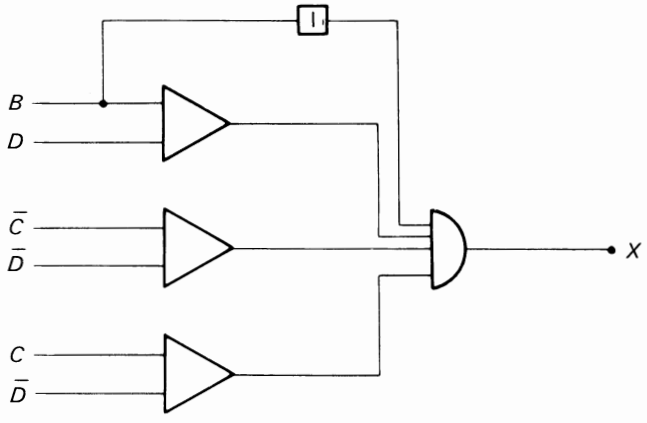


Fig. 61.

Calculemos primeramente X. Obtenemos según la figura 61:

$$X = \bar{B} + BD + \bar{C}\bar{D} + C\bar{D}\,.$$

Trasladamos esta función al diagrama de Karnaugh (Fig. 62).

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Fig. 62. $X = 1$.

Observaremos con asombro que todas las cuadrículas del diagrama están ocupadas con "1". Lo que quiere decir que cualesquiera que sean las combinaciones de las variables de la entrada, la salida es siempre 1. El esquema equivalente es el de la Fig. 63. En este caso se observará que el método del diagrama de Karnaugh permite obtener mucho más rápidamente la solución. Recordemos que en el capítulo 1 este ejemplo se trata únicamente mediante los teoremas del álgebra de Boole.

Fig. 63. *Esquema equivalente al de la figura* 61.

3.7 Circuitos equivalentes bajo la forma de un producto de varias sumas

Hasta aquí hemos expresado la función X bajo la forma de una suma de productos que hemos hecho mínima mediante las técnicas del diagrama de Karnaugh.

El circuito equivalente se compone entonces de varios circuitos Y reunidos en un circuito O. Podemos igualmente representar X bajo la forma de un producto de varias sumas. El circuito equivalente estará compuesto de varios circuitos O reunidos en un circuito Y. Para obtener este circuito calculemos $\overline{X}$ mediante el diagrama de Karnaugh. Si elegimos convenientemente nuestros bucles, $\overline{X}$ será óptimo. Se obtiene X invirtiendo la expresión de $\overline{X}$ por medio del teorema de De Morgan.

Esto equivale a decir que existen siempre dos versiones simplificadas de un mismo circuito. Habrá que comprobar siempre cuál es el circuito más simple y económico.

Ejemplo 1. *Representar el circuito equivalente del diagrama de Karnaugh de la figura 64 bajo la forma de una suma de productos o de un producto de sumas.*

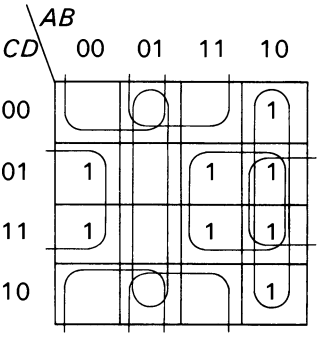


Fig. 64. $X = A\overline{B} + AD + \overline{B}D$
$\overline{X} = \overline{A}B + B\overline{D} + \overline{A}\overline{D}$.

Haciendo los bucles como en la figura 64, obtenemos:

$$X = A\overline{B} + AD + \overline{B}D\,,$$

que corresponde al esquema equivalente de la figura 65. Equivale a una suma de productos. Haciendo los bucles de las cuadrículas vacías de la figura 64, se obtiene:

$$\overline{X} = \overline{A}B + B\overline{D} + \overline{A}\overline{D}$$

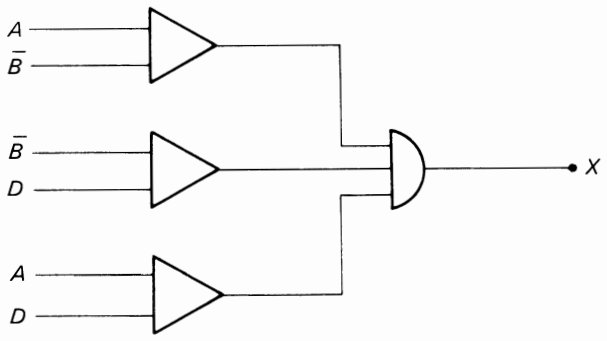


Fig. 65. *Esquema equivalente al diagrama de la figura* 64.

y volviendo a X:

$$X = \overline{\overline{X}} = \overline{\overline{A}B + B\overline{D} + \overline{A}\overline{D}}\,.$$

es decir, aplicando el teorema de De Morgan:

$$X = (A + \overline{B})\,(\overline{B} + D)\,(A + D)\,.$$

La función X se expresa mediante un producto de sumas. El circuito equivalente es el de la figura 66:

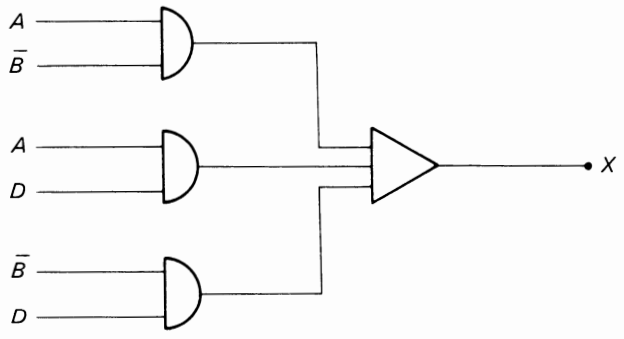


Fig. 66. *Esquema equivalente al de la figura* 65.

Se observará que las dos versiones son equivalentes desde el punto de vista económico (número de circuitos).

Ejemplo 2. *El mismo problema que para el ejemplo* 1, *pero para el diagrama de Karnaugh de la figura 67:*

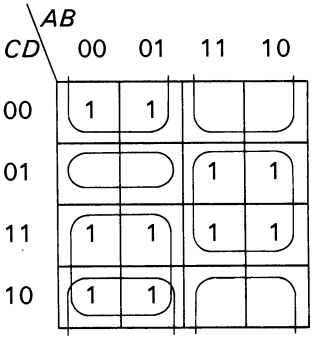


Fig. 67. $X = \bar{A}\bar{D} + AD + \bar{A}C$
$\bar{X} = (\bar{A} + D)\,(A + C + \bar{D})$.

De la figura 67, obtenemos:

$$X = \bar{A}\bar{D} + AD + \bar{A}C\,.$$

lo que corresponde al esquema equivalente de la figura 68:

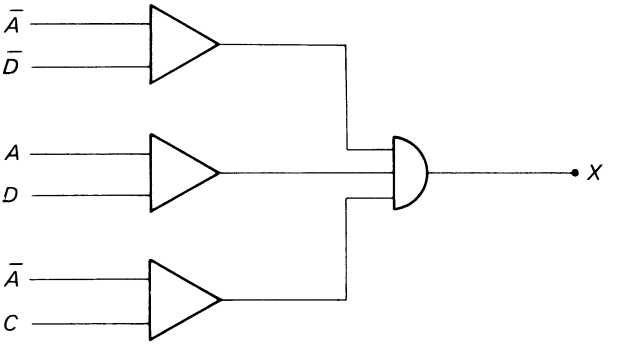


Fig. 68. *Esquema equivalente al diagrama de la figura* 67.

Calculemos $\bar{X}$ de la figura 67:

$$\bar{X} = A\bar{D} + \bar{A}\bar{C}D$$

y:

$$X = \bar{\bar{X}} = \overline{A\bar{D} + \bar{A}\bar{C}D}\,.$$

De acuerdo con el teorema de De Morgan:

$$X = (\bar{A} + D)\,(A + C + \bar{D})\,,$$

que corresponde al esquema de la figura 69. Se observará que este último esquema es más económico que el de la figura 68 (siete entradas en lugar de nueve).

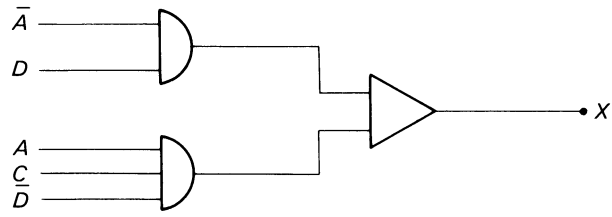


Fig. 69. *Esquema equivalente al de la figura* 68.

Ejemplo 3. *El mismo problema que para los dos ejemplos anteriores, pero para el diagrama de Karnaugh de la figura 70:*

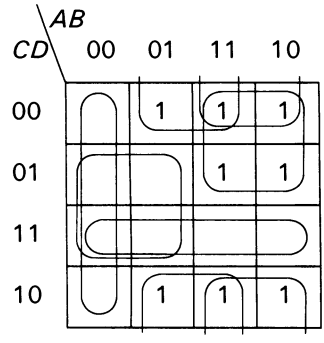


Fig. 70. $X = A\bar{C} + A\bar{D} + B\bar{D}$
$\bar{X} = \bar{A}\bar{B} + CD + \bar{A}D.$

Obtenemos, siguiendo el mismo método que en los casos anteriores:

$$X = A\bar{C} + A\bar{D} + B\bar{D},$$

que corresponde al esquema de la figura 71:

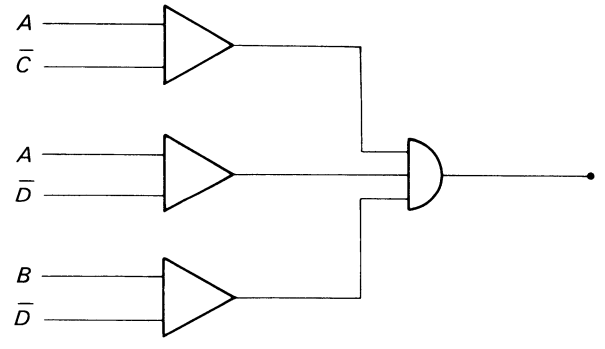


Fig. 71. *Esquema equivalente al diagrama de la figura* 70.

Por otro lado:

$$\bar{X} = \bar{A}\bar{B} + CD + \bar{A}D ,$$

de donde:

$$X = \overline{\bar{A}\bar{B} + CD + \bar{A}D} ,$$

es decir:

$$X = (A + B)(\bar{C} + \bar{D})(A + \bar{D}) ;$$

lo que corresponde al esquema de la figura 72. Se observará que los dos esquemas son equivalentes desde el punto de vista económico.

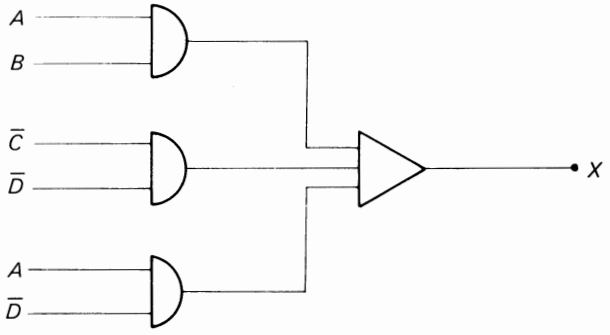


Fig. 72. *Esquema equivalente al de la figura* 71.

Ejemplo 4. *Lo mismo que en los ejemplos anteriores pero para el diagrama de Karnaugh de la figura 73:*

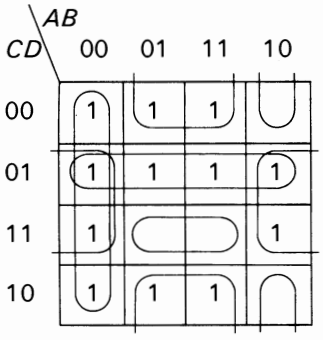


Fig. 73. $X = \bar{A}\bar{B} + \bar{C}D + B\bar{D} + \bar{B}D$
$\bar{X} = BCD + A\bar{B}\bar{D}.$

Siguiendo el mismo método que anteriormente, obtenemos:

$$X = \bar{A}\bar{B} + \bar{C}D + B\bar{D} + \bar{B}D ,$$

que corresponde al esquema de la figura 74:

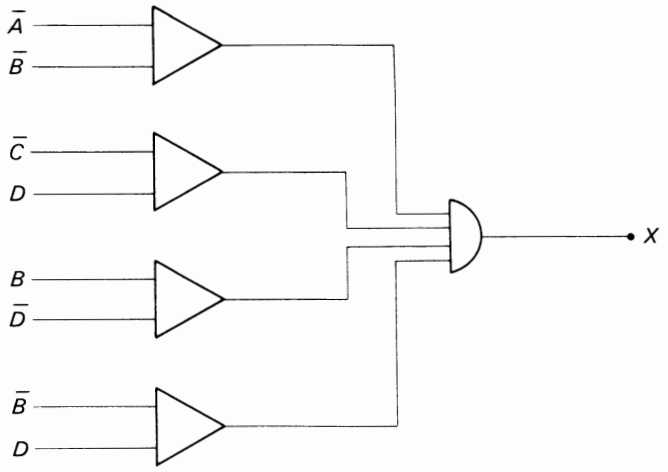


Fig. 74. *Esquema equivalente al diagrama de la figura* 72.

Igualmente obtenemos:

$$X = (\overline{B} + \overline{C} + \overline{D})\,(\overline{A} + B + D)\,,$$

que corresponde al esquema de la figura 75. Se observará, que este esquema es más simple que el de la figura 74, y por tanto más económico.

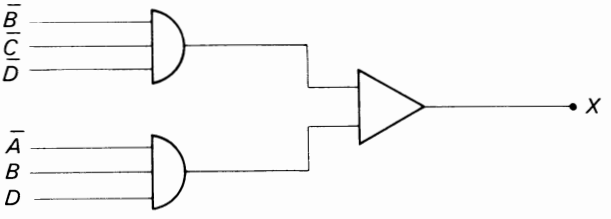


Fig. 75. *Esquema equivalente al de la figura* 73.

3.8 Factorización de uno o varios términos

Una vez obtenida la función X más simple mediante el diagrama de Karnaugh, bajo la forma de una suma de productos, se verificará si es posible poner en forma de producto uno o varios términos, en este caso se puede simplificar ventajosamente el circuito equivalente.

Volvamos al ejemplo 1 del párrafo 3.6. Habíamos encontrado una solución simplificada:

$$X = A\overline{B} + AD + \overline{B}D\,.$$

y sacando $\overline{B}$ factor común obtenemos:

$$X = \overline{B}(A + D) + AD ,$$

lo que corresponde al esquema de la figura 76. Se observará que este esquema es más simple que el de la figura 65.

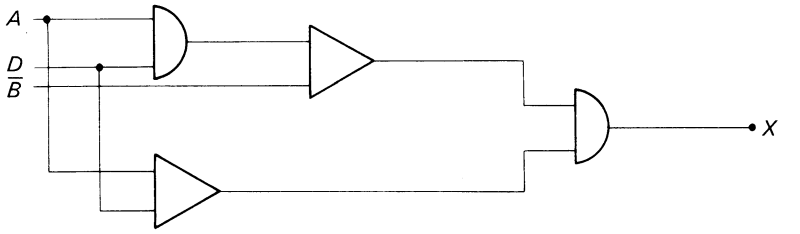


Fig. 76. *Esquema equivalente al de la figura* 65.

En el ejemplo 3 del mismo párrafo, habíamos hallado:

$$X = A\overline{C} + A\overline{D} + B\overline{D} .$$

Sacando $\overline{D}$ factor común obtenemos:

$$X = A\overline{C} + \overline{D}(A + B) .$$

El esquema equivalente es el de la figura 77. Una vez más este esquema es más simple que el de la figura 71. Se observará, por el contrario, que el ejemplo 2 se puede simplificar más. Veremos que el sacar factor común en algunos términos origina en ocasiones una complejidad suplementaria.

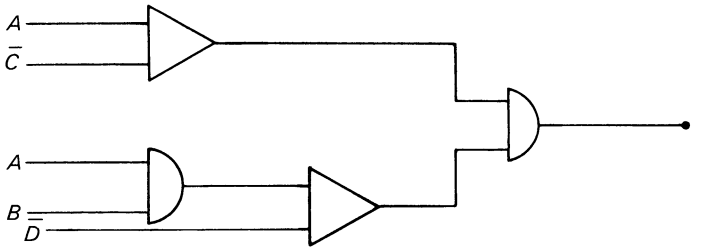


Fig. 77. *Esquema equivalente al de la figura* 70.

4. Condiciones prohibidas

El lector habrá observado la elegancia del método utilizado hasta aquí. Pero la mayor potencia de los diagramas de Karnaugh reside sobre todo en la posibilidad de incorporar a los diagramas las "condiciones prohibidas" y efectuar los cálculos y simplificaciones teniendo en cuenta estas condiciones. Entendemos por condiciones prohibidas las combinaciones de variables que no se producen jamás. Si una combinación no se produce nunca, la función X es independiente de esta combinación y podemos atribuirle el valor 1 ó el 0 en el diagrama de Karnaugh para obtener la solución más simple. Esta combinación prohibida se representa por ϕ (0 y 1 superpuestos).

A lo largo de la simplificación se le atribuirá el valor 1 ó el 0 para obtener el mínimo de bucles y los más grandes.

Ejemplo 1. *Consideremos una función de cuatro variables "booleanas" $X = f(A, B, C, D)$ tal que $X = 1$ si:*

$$\begin{array}{llll} A=0, & B=1, & C=0, & D=0, \\ A=1, & B=1, & C=0, & D=0, \\ A=1, & B=0, & C=0, & D=1, \\ A=1, & B=0, & C=1, & D=1. \end{array}$$

Las siguientes condiciones no se utilizan (no se producen jamás):

$$\begin{array}{llll} A=0, & B=0, & C=0, & D=0, \\ A=1, & B=0, & C=0, & D=0, \\ A=0, & B=0, & C=0, & D=1, \\ A=0, & B=0, & C=1, & D=1, \\ A=0, & B=1, & C=0, & D=1, \\ A=1, & B=1, & C=0, & D=1, \\ A=0, & B=1, & C=1, & D=0, \\ A=1, & B=1, & C=1, & D=0. \end{array}$$

Para las otras condiciones $X = 0$. Se puede determinar la solución más simple de X. Para el esquema equivalente.

Llamemos ϕ a las condiciones que no se utilizan. Pueden tomar indiferentemente los valores 1 ó 0. El diagrama de Karnaugh correspondiente al enunciado es el de la figura 78.

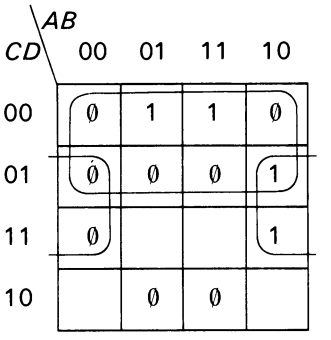


Fig. 78. $X = \bar{C} + \bar{B}D$.

Para simplificar el valor de X, se buscará, como anteriormente, realizar los bucles mayores posibles. Así las ϕ englobadas en los bucles de la figura 78, tienen el valor 1, los otros el valor 0. Se obtiene así la solución más simple para X:

$$X = \bar{C} + \bar{B}D\,,$$

que corresponde al esquema equivalente de la figura 79:

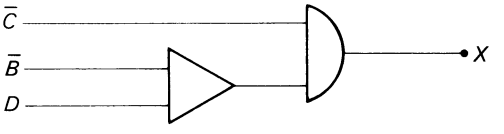


Fig. 79. *Esquema equivalente al diagrama de la figura* 78.

Se han incorporado así las condiciones no utilizadas para la simplificación del circuito equivalente. Se observará que si se produjesen las condiciones prohibidas incorporadas en los bucles, darían un 1 en la salida.

Ejemplo 2. *Consideremos una función de cuatro variables* $X = f(A, B, C, D)$ *tal que* $X = 1$ *si:*

$$A = 0\,, \quad B = 1\,, \quad C = 0\,, \quad D = 1\,,$$
$$A = 0\,, \quad B = 1\,, \quad C = 1\,, \quad D = 1\,.$$

Están prohibidas las siguientes condiciones:

$$A = 1\,, \quad B = 0\,, \quad C = 0\,, \quad D = 0\,,$$
$$A = 1\,, \quad B = 1\,, \quad C = 0\,, \quad D = 1\,,$$
$$A = 1\,, \quad B = 1\,, \quad C = 1\,, \quad D = 1\,.$$

La función X es nula para las restantes combinaciones. Se pide determinar el circuito equivalente más simple.

El diagrama de Karnaugh es el de la figura 80, donde se llevan todas las condiciones del enunciado. Para realizar el mayor bucle posible, se atribuye el valor 1 a dos condiciones no

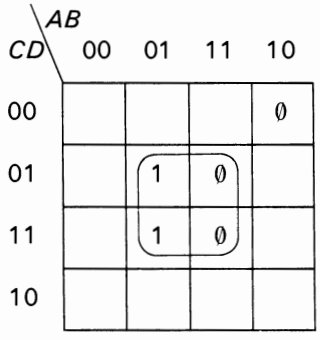


Fig. 80. $X = BD$.

utilizadas. En estas condiciones la solución más simple es:

$$X = BD$$

que se realiza mediante un simple circuito Y (Fig. 81):

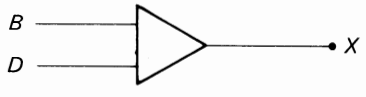


Fig. 81. *Esquema equivalente al diagrama de la figura* 80.

Ejemplo 3. *Sea una función de cuatro variables, X = f (A, B, C, D). La salida será 1 cada vez que sólo dos variables sean iguales a 1. No hay que preocuparse de la salida cuando tres o cuatro variables son iguales a 1. Cual es el circuito más simple que realiza esta función.*

La función posee cuatro variables, luego $2^4 = 16$ combinaciones. La tabla de verdad es la siguiente. Llevemos X al diagrama de Karnaugh, figura 82:

A	*B*	*C*	*D*	*X*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	∅
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	∅
1	1	0	0	1
1	1	0	1	∅
1	1	1	0	∅
1	1	1	1	∅

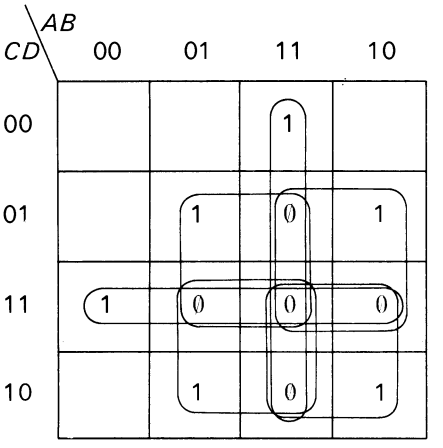


Fig. 82. $X = AB + CD + BC + BD + AD + AC$.

Tendremos seis bucles con cuatro cuadrículas adyacentes, en que todos los ϕ son iguales a 1 en esta aplicación. La solución es:

$$X = AB + CD + BC + BD + AD + AC\,.$$

Podemos hacer más simple esta expresión sacando factor común A y D.

$$X = A(B + C + D) + D(B + C) + BC .$$

El esquema equivalente que realiza las condiciones del enunciado es el de la figura 83. Se observará que las condiciones para las cuales tres o cuatro variables son iguales a (ϕ) se han englobado en los bucles de simplificación: se les ha atribuido arbitrariamente el valor 1. Para estas combinaciones que se producen a la entrada, la salida tomará el valor 1, pero como no nos importa la salida para estas combinaciones, ellas pueden tomar el valor 1.

No intervendrán en el resultado final. Se puede suponer que la salida X estará bifurcada a un dispositivo que sólo se sensibilizará en las combinaciones que aportan dos y solamente dos 1.

Este caso es diferente del ejemplo siguiente, donde un determinado número de combinaciones son prohibidas, es decir no se producen nunca.

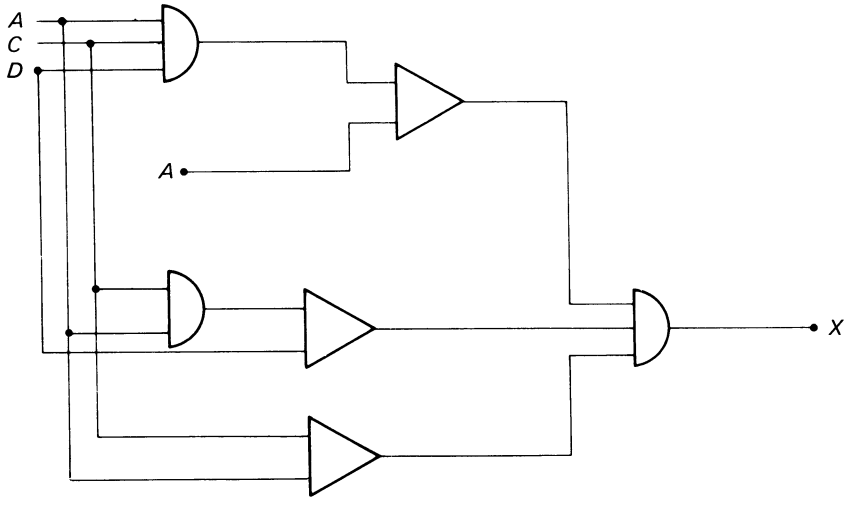


Fig. 83. *Esquema equivalente al diagrama de la figura* 82.

Ejemplo 4. *Determinar cuatro circuitos separados en cuatro entradas, tales que las combinaciones de las salidas representan el complemento a 9 del número decimal codificado en BCD presente a la entrada.*

De acuerdo con el enunciado del problema hay $2^4 = 16$ combinaciones de las variables de entrada. No preocupan las combinaciones binarias de las cifras 10 a 15 que se supone no se producen. Llamemos W, X, Y y Z las salidas respectivas de cada uno de los cuatro circuitos separados. La tabla de verdad es la siguiente:

Decimal	*A*	*B*	*C*	*D*	*W*	*X*	*Y*	*Z*
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	0	1	1
7	0	1	1	1	0	0	1	0
8	1	0	0	0	0	0	0	1
9	1	0	0	1	0	0	0	0
10	1	0	1	0	∅	∅	∅	∅
11	1	0	1	1	∅	∅	∅	∅
12	1	1	0	0	∅	∅	∅	∅
13	1	1	0	1	∅	∅	∅	∅
14	1	1	1	0	∅	∅	∅	∅
15	1	1	1	1	∅	∅	∅	∅

A continuación se establece los diagramas de Karnaugh para *W, X, Y* y *Z* (Figs. 84 a 86).

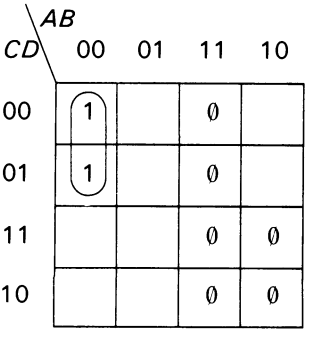


Fig. 84. $W = \bar{A}\bar{B}\bar{C}$

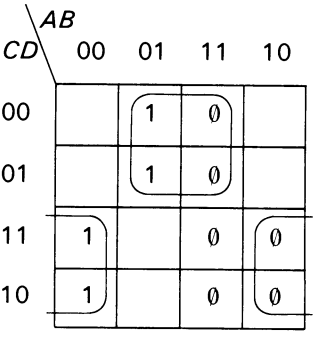


Fig. 85. $X = B\bar{C} + \bar{B}C$.

CD \ AB	00	01	11	10
00			∅	
01			∅	
11	1	1	∅	∅
10	1	1	∅	∅

Fig. 86. $Y = C$.

CD \ AB	00	01	11	10
00	1	1	∅	1
01			∅	
11			∅	∅
10	1	1	∅	∅

Fig. 87. $Z = \overline{D}$.

Obtenemos los siguientes resultados:

$$W = \overline{A}\overline{B}\overline{C},$$
$$X = B\overline{C} + \overline{B}C = B \oplus C,$$
$$Y = C,$$
$$Z = \overline{D};$$

de aquí el esquema de la figura 88:

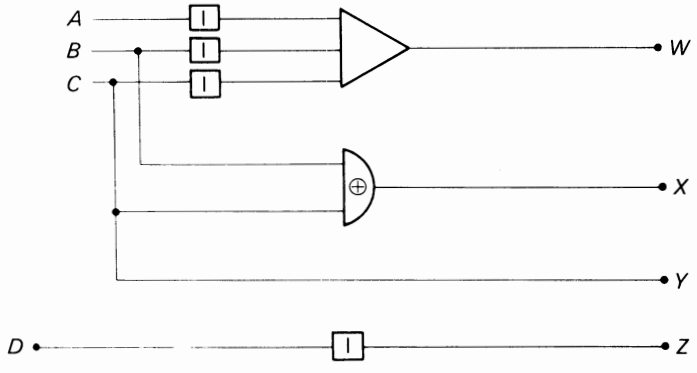


Fig. 88. *Circuito de complemento a* 9.

4.1 Inversión de una expresión que contiene dos condiciones prohibidas

Para obtener la inversa de una función de conmutación $\overline{X}$ conociendo su valor X en el diagrama de Karnaugh, se procederá de la siguiente forma:

- rodear con bucles los ceros del diagrama de Karnaugh, incorporando algunas condiciones prohibidas para obtener bucles mayores;
- el valor de bucles debe ser mínimo, lo que lleva a atribuir el valor $\phi = 1$ a algunas condiciones prohibidas.

En los casos complicados se calculará desde el principio X de la misma forma que hemos hecho hasta aquí, después se rodearán con bucles las cuadrículas vacías y los valores prohibidos que no se han estilizado en el cálculo de X.

Ejemplo 1. *Hallar el esquema equivalente de la función $\overline{X}$, conociendo el diagrama de Karnaugh de la función X (Fig. 89).*

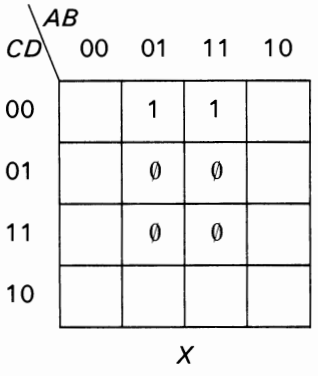


Fig. 89. *Diagrama de Karnaugh de la función X del problema* 1.

Se calculará inicialmente X (Fig. 90) después $\overline{X}$ incorporando en los bucles las condiciones prohibidas que no se han utilizado para el cálculo de X.

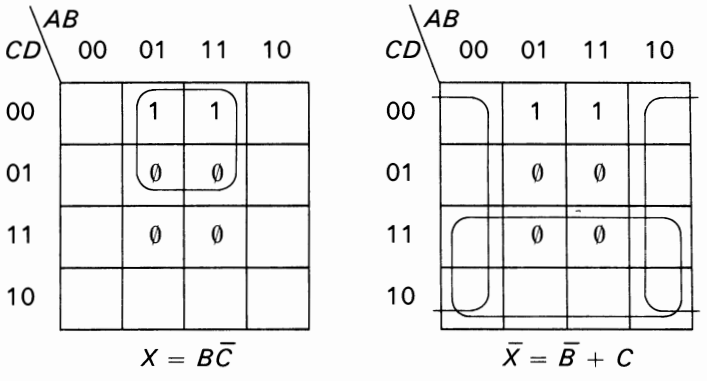


Fig. 90. *Cálculo de X y $\overline{X}$ del problema* 1.

La lectura de los diagramas da:

$$X = B\overline{C}$$
$$\overline{X} = \overline{B} + C$$

y el esquema equivalente es el de la figura 91. La relación puede efectuarse igualmente en dos circuitos separados X y $\overline{X}$. Normalmente se prefiere la solución de la figura 91, que da simultáneamente X y $\overline{X}$.

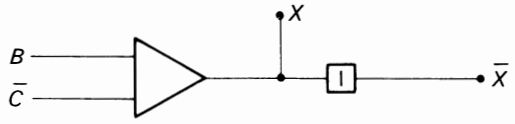


Fig. 91. *Esquema lógico equivalente del problema* 1.

Ejemplo 2. *El mismo enunciado que para el problema* 1, *pero con el diagrama de la figura 92.*

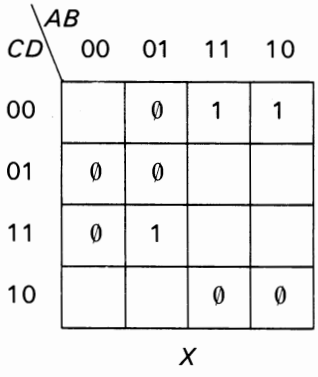


Fig. 92. *Diagrama de Karnaugh de la función X del problema* 2.

El cálculo de X y $\bar{X}$ se obtiene mediante los diagramas de las figuras 93a y *b.*

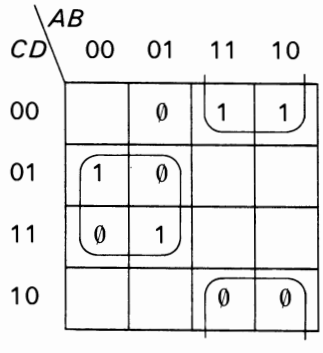


Fig. 93a. $X = \bar{A}D + A\bar{D}$

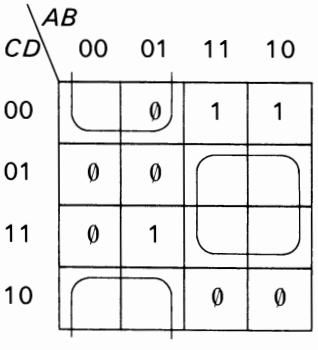


Fig. 93b. $\bar{X} = AD + \bar{A}\bar{D}$

$$X = \bar{A}D + A\bar{D} = A \oplus D\,,$$
$$\bar{X} = AD + \bar{A}\bar{D} = A \odot D\,,$$

y el circuito lógico es el de la figura 94.

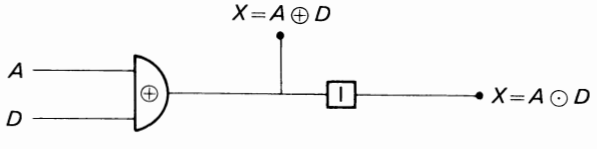


Fig. 94. *Circuito lógico del problema* 2.

5. Circuitos de múltiples salidas

Consideramos dos funciones "booleanas" de cuatro variables: A, B, C, D, bajo la forma de suma de productos. Si uno de los productos es común a las dos funciones, se realizará un solo producto, que se utilizará dos veces para obtener las dos funciones. En los diagramas de Karnaugh las dos funciones se traducirán por la existencia de dos bucles idénticos.

Inversamente cada vez que se descubre sobre dos o más diagramas de Karnaugh, la existencia de bucles idénticos, se estará en presencia de una fracción de circuito común a las correspondientes funciones.

Ejemplo 1. *Consideremos los dos diagramas de Karnaugh, de las figuras 95 y 96, donde se llevan separadamente las funciones X e Y.*

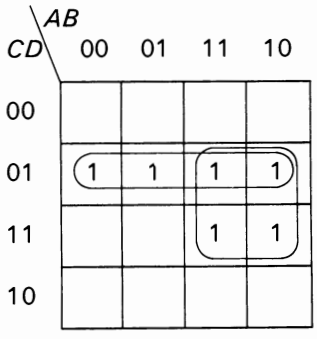


Fig. 95. *X.*

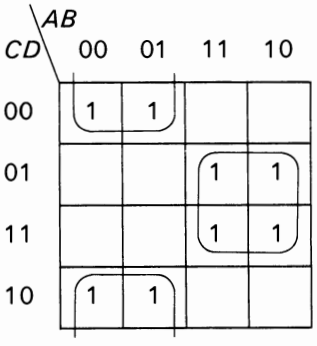


Fig. 96. *Y.*

Se observará la existencia de dos bucles idénticos en cada uno de los diagramas, lo que equivale a la existencia de un término idéntico en las expresiones de X e Y.

En efecto, obtenemos:

$$X = \bar{C}D + AD$$
$$Y = A\bar{D} + AD\,.$$

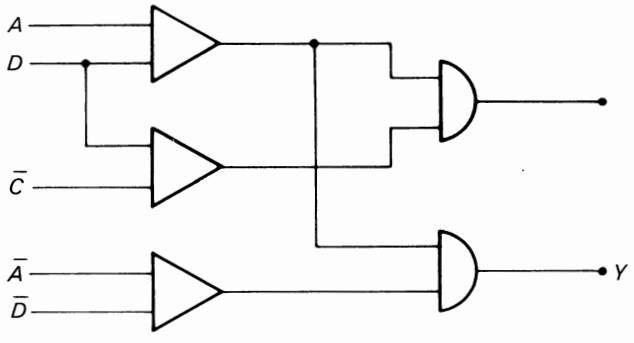


Fig. 97. *Esquema equivalente a los diagramas de las figuras* 95 *y* 96.

El término AD es común a las dos expresiones. Se realizará un único circuito AD que se utilizará dos veces para la realización de las funciones X e Y, el esquema equivalente es el de la figura 97.

Ejemplo 2. *Realizar el mejor circuito que, teniendo tres salidas, X, Y y Z, responda a los siguientes diagramas de Karnaugh (Figs. 98, 99 y 100).*

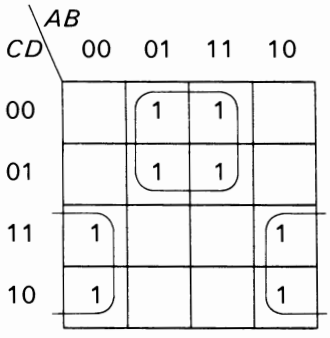


Fig. 98. *X*.

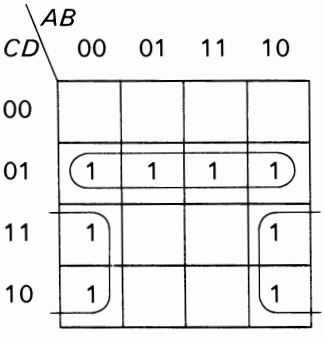


Fig. 99. *Y*.

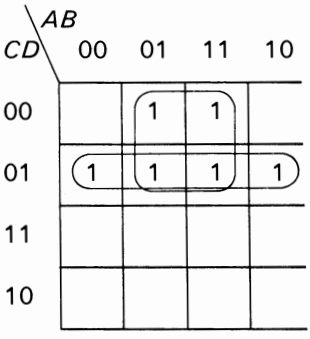


Fig. 100. *Z*.

Observamos la existencia de tres bucles idénticos. Hay un circuito común a las funciones X e Y, X y Z, y X y Z.

Tenemos:

$$X = B\bar{C} + \bar{B}C\,,$$
$$Y = \bar{C}D + \bar{B}C\,,$$
$$Z = B\bar{C} + \bar{C}D\,.$$

$\bar{B}C$ es común a X e Y, $\bar{C}D$ a Y y Z y $B\bar{C}$ a X y Z. El circuito equivalente más simple es el de la figura 101.

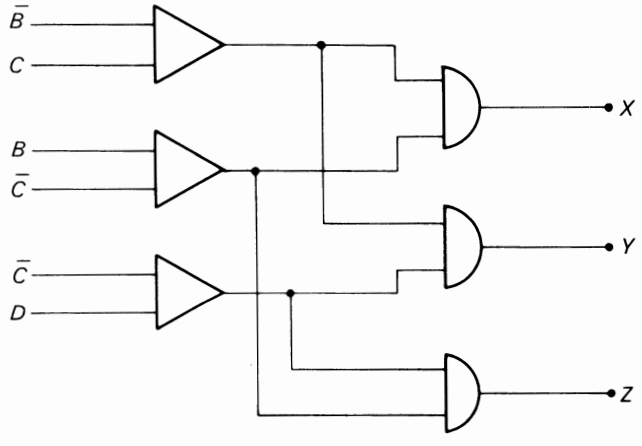


Fig. 101. *Esquema equivalente a los tres diagramas de las figuras* 98, 99 *y* 100.

Ejemplo 3. *Determinar el mejor circuito con dos salidas X e Y que responda a los diagramas de Karnaugh de las figuras* 102 *y* 103.

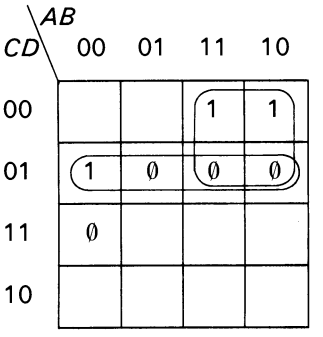


Fig. 102. *X.*

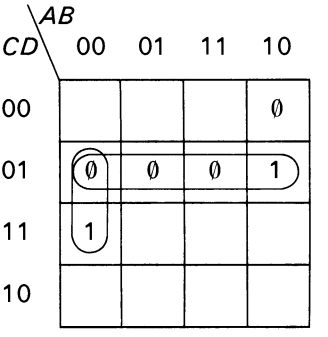


Fig. 103. *Y.*

Se observará en estos diagramas la existencia de dos bucles comunes: cuyos circuitos serán comunes a *X* e *Y*. En efecto, la lectura de los diagramas da:

$$X = A\bar{C} + \bar{C}D,$$
$$Y = \bar{A}\bar{B}D + \bar{C}D.$$

El término $\bar{C}D$ es común a las dos funciones *X* e *Y*. El esquema equivalente es el de la figura 104.

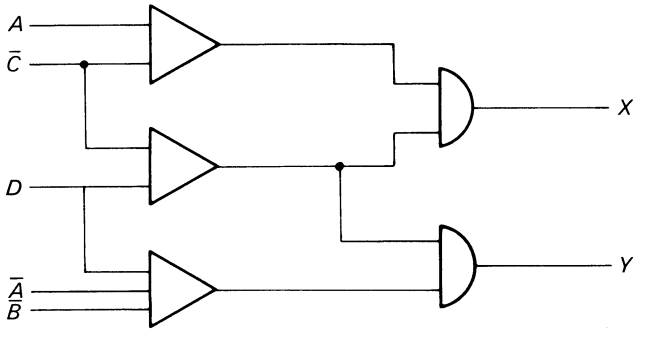


Fig. 104. *Esquema equivalente a los diagramas de las figuras* 102 *y* 103.

Realización de los circuitos básculas *T*, *J-K*, *R-S* y *R-S-T* mediante circuitos lógicos

Hemos estudiado en el capítulo anterior el funcionamiento estático de los circuitos de básculas *T*, *R-S* y *R-S-T*.

Hemos obtenido sus funciones lógicas realizadas entre el estado anterior de la báscula y las órdenes aplicadas en sus distintas entradas. Vamos a demostrar en este párrafo que las funciones lógicas de los circuitos báscula pueden realizarse mediante circuitos lógicos convenientemente asociados entre ellos. Se preferirá sin embargo la utilización de los circuitos báscula para la realización de estos esquemas.

Caso del circuito báscula T

Este circuito se ha estudiado en el capítulo 1. Hemos visto que sus salidas en el instante $t + 1$, Q_{t+1} y $\overline{Q_{t+1}}$ realizan las siguientes funciones lógicas entre su estado anterior Q_t y su entrada T:

$$\begin{aligned} Q_{t+1} &= Q_t \oplus T \\ \overline{Q_{t+1}} &= Q_t \odot T . \end{aligned} \tag{1}$$

El circuito T realiza en su salida un O exclusivo entre su estado anterior y su entrada T y sobre su salida complementaria la coincidencia entre su estado anterior y su entrada T. *Una vez que ha pasado al estado* Q_{t+1} *éste se convierte en el nuevo estado anterior.*

El circuito T puede representarse, según el caso, por un circuito O exclusivo o un circuito de coincidencia cuya salida se enlaza con la entrada (Fig. 105).

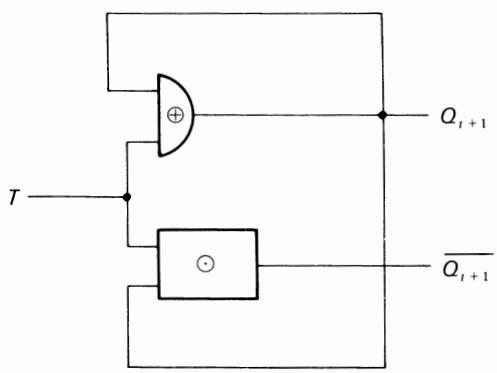


Fig. 105. *Realización de un circuito báscula T con n circuitos lógicos.*

Caso del circuito báscula J-K

El circuito *J-K* posee dos entradas J y K. K asegura la puesta a cero del circuito J y su puesta a 1. Cambia de estado cualquiera que sea su estado anterior cuando las dos entradas J y K están en 1. La función lógica de este circuito se ha estudiado en el capítulo 1. Hemos hallado:

$$\begin{aligned} Q_{t+1} &= J\overline{Q_t} + \overline{K}Q_t \\ \overline{Q_{t+1}} &= \overline{J}\overline{Q_t} + KQ_t . \end{aligned} \tag{2}$$

Lo que corresponde a la realización de dos circuitos separados independientes el uno del otro.

Si se desea realizar un esquema equivalente *entregando simultáneamente* sobre una *misma* salida Q_{t+1} y $\overline{Q_{t+1}}$ se calculará primeramente Q_{t+1} después se sumará su inverso (Fig. 106).

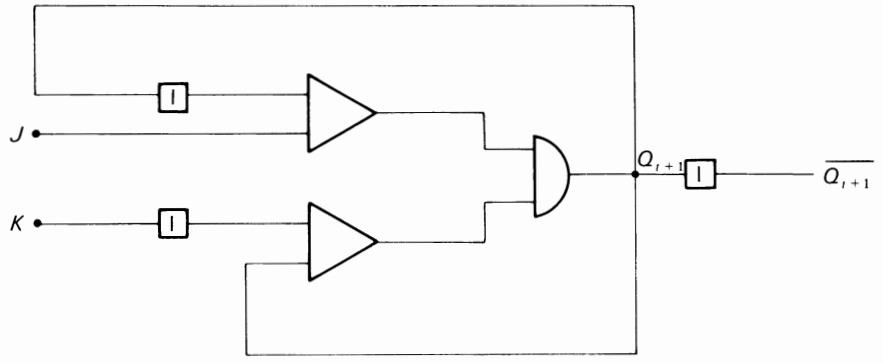


Fig. 106. *Realización con circuitos lógicos, de un circuito basculante J-K.*

Caso del circuito báscula R-S

El circuito *R-S* se ha estudiado en el capítulo 1 mediante el álgebra de Boole. Aquí nos proponemos estudiarlo mediante el diagrama de Karnaugh. Se recordará que la entrada *R* está destinada a poner a cero (Reset) y la entrada *S* (Set) a la puesta en 1 de la báscula.

No pueden producirse dos 1 simultáneamente en las entradas *R* y *S:* ésta es una condición prohibida. La tabla de verdad se da en la figura 107.

Q_t	R	S	Q_{t+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	∅
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	∅

Fig. 107. *Tabla de verdad del circuito báscula RS.*

El diagrama de Karnaugh es el de la figura 108. Si *se quieren* calcular *separadamente* las dos funciones Q_{t+1} y $\overline{Q_{t+1}}$, es decir realizar dos funciones *independien-*

tes se englobarán las condiciones ϕ en los bucles para el cálculo de Q_{t+1} y $\overline{Q_{t+1}}$. En el primer caso se atribuye a ϕ el valor 1, en el segundo caso el valor 0:

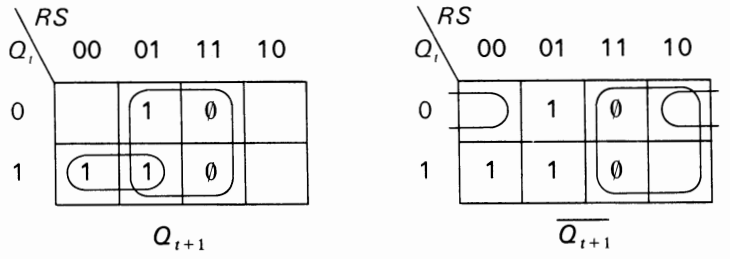


Fig. 108. *Diagramas de Karnaugh para el cálculo de* Q_{t+1} *y* $\overline{Q_{t+1}}$.

La lectura de los diagramas da:

$$\begin{aligned} Q_{t+1} &= S + \overline{R}\, Q_t \\ \overline{Q_{t+1}} &= R + \overline{S}\, \overline{Q_t}\,. \end{aligned} \tag{3}$$

Estas dos operaciones son *distintas* e *independientes*, es decir $\overline{Q_{t+1}}$ no depende de Q_{t+1} en lo que respecta a las condiciones prohibidas. Son las ecuaciones que hemos obtenido en el capítulo 1. Los circuitos lógicos equivalentes que simulan las salidas de la báscula se dan en la figura 109.

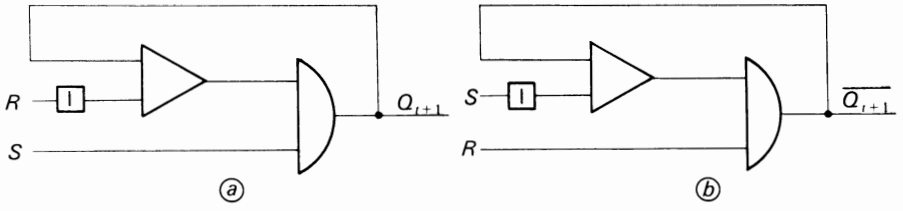


Fig. 109. Esquemas equivalentes de un circuito báscula *RS*, ⓐ y ⓑ. *Soluciones separadas de* Q_{t+1} *y* $\overline{Q_{t+1}}$ (ⓐ : $\emptyset = 1$, ⓑ : $\emptyset = 0$).

Por el contrario si deseamos obtener en una *salida única* Q_{t+1} y $\overline{Q_{t+1}}$ hemos de disponer de un inversor a la salida del esquema ⓐ. En este caso obtenemos, aplicando el teorema de De Morgan:

$$\overline{Q_{t+1}} = \overline{S + \overline{R} Q_t} = \overline{S}(R + \overline{Q_t})\,. \tag{4}$$

Observemos que este valor difiere del obtenido anteriormente en (3). En efecto, esta vez $\overline{Q_{t+1}}$ depende de Q_{t+1} y las condiciones prohibidas conservan los valores $\phi = 1$ que se les han atribuido para el cálculo de Q_{t+1}.

No pueden englobarse en los bucles para el cálculo de $\overline{Q_{t+1}}$ (Fig. 110).

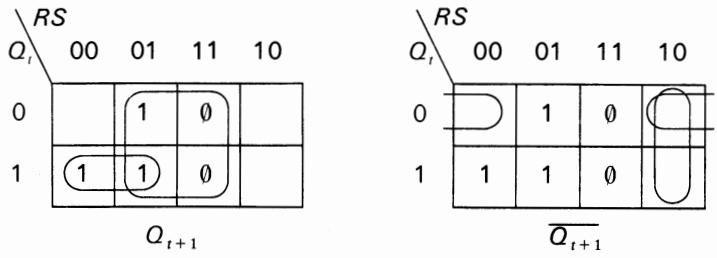


Fig. 110. *Diagramas de Karnaugh para el cálculo simultáneo de* Q_{t+1} *y* $\overline{Q_{t+1}}$.

El esquema equivalente es el de la figura 111:

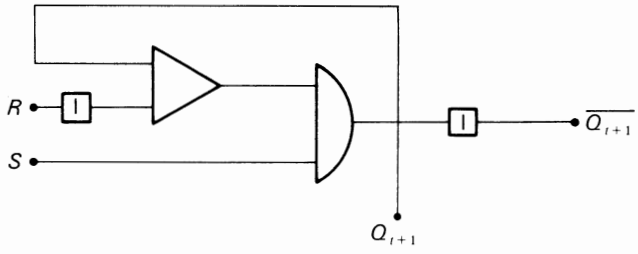


Fig. 111. *Solución simultánea de* Q_{t+1} *y* $\overline{Q_{t+1}}$.

Circuito báscula R-S-T

El caso del circuito báscula *R-S-T* corresponde al funcionamiento superpuesto de la báscula *R-S* y *T*. No se podrán aplicar nunca dos o tres unos sobre las entradas *R*, *S*, *T*. La tabla de verdad es la de la figura 111. Hay 16 combinaciones con 8 condiciones prohibidas. El diagrama de Karnaugh es el de la figura 113. La lectura de este diagrama da:

$$Q_{t+1} = S + \overline{Q}_t\, T + Q_t\, \overline{R}\,\overline{T}\,. \qquad (5)$$

Q_t	R	S	T	Q_{t+1}
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	∅
0	1	0	0	0
0	1	0	1	∅
0	1	1	0	∅
0	1	1	1	∅
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	∅
1	1	0	0	0
1	1	0	1	∅
1	1	1	0	∅
1	1	1	1	∅

Fig. 112. *Tabla de verdad del circuito R-S-T.*

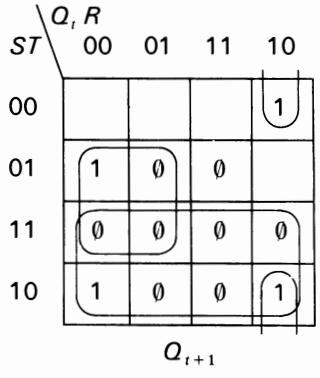


Fig. 113. *Diagrama de Karnaugh del circuito R-S-T.*

Si se calcula *separadamente* la salida $\overline{Q_{t+1}}$, obtenemos el diagrama de la figura 114, donde se han incorporado las condiciones prohibidas en los bucles ($\emptyset = 0$).

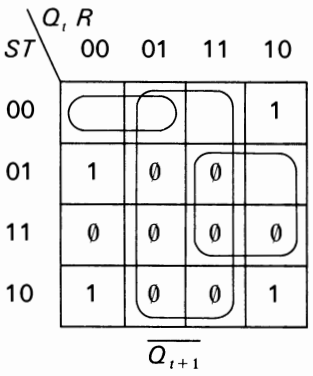


Fig. 114. *Diagrama de Karnaugh para el cálculo de* $\overline{Q_{t+1}}$ *en el caso de una solución independiente de* Q_{t+1}.

La lectura del diagrama de Karnaugh da:

$$\overline{Q_{t+1}} = R + TQ_t + \overline{STQ_t}\,. \tag{6}$$

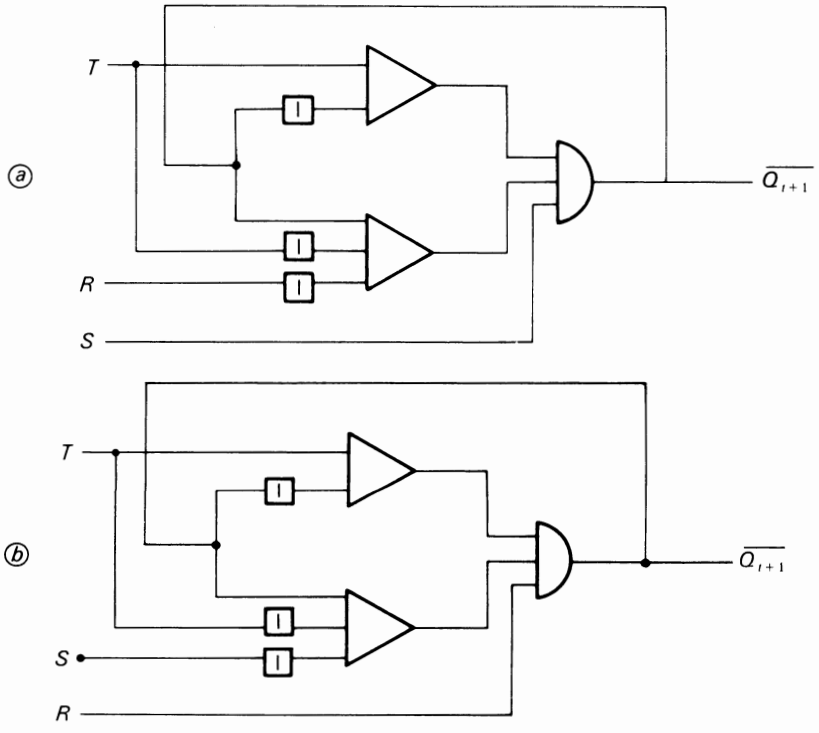


Fig. 115. *Realización de los esquemas* Q_{t+1} ⓐ *et* $\overline{Q_{t+1}}$ ⓑ *(soluciones independientes).*

Los esquemas equivalentes se dan en la figura 115. En el caso que se deseen obtener Q_{t+1} y $\overline{Q_{t+1}}$ en una misma salida, se calculará primeramente Q_{t+1} y después se colocará un inversor a la salida. En este caso se efectuará una solución simultánea de Q_{t+1} y $\overline{Q_{t+1}}$. Los valores prohibidos, rodeados de un bucle para el cálculo de Q_{t+1}, conservarán sus valores 1 en el cálculo de $\overline{Q_{t+1}}$ y no podrán rodearse de bucles en el diagrama $\overline{Q_{t+1}}$ (Fig. 116).

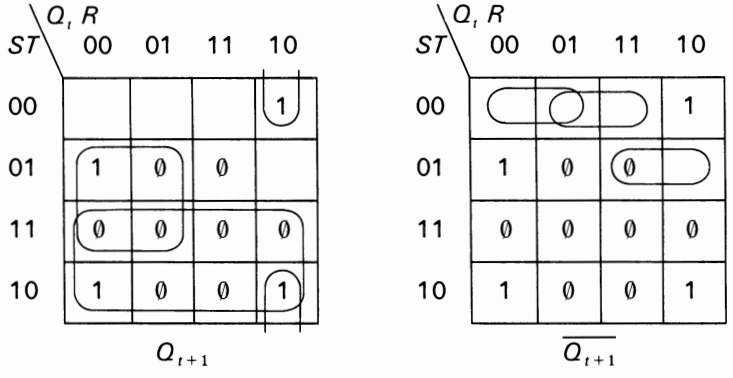


Fig. 116. *Diagramas de Karnaugh del circuito báscula R-S-T en el caso de una solución simultánea de* Q_{t+1} *y* $\overline{Q_{t+1}}$.

Obtenemos:

$$\overline{Q_{t+1}} = \overline{S}(Q_t\, T + \overline{Q}_t\, \overline{T} + R\overline{T})$$

valor distinto del obtenido anteriormente, pero igual al que se obtendría invirtiendo la expresión simplificada de Q_{t+1}. El esquema equivalente es el de la figura 117.

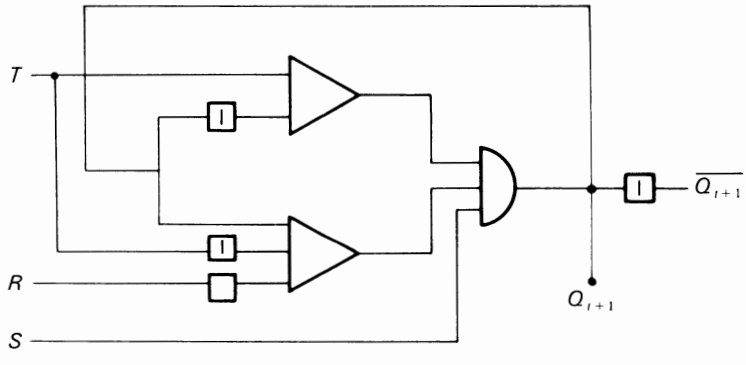


Fig. 117. *Esquema equivalente de un circuito báscula R-S-T (solución simultánea de* Q_{t+1} *y* $\overline{Q_{t+1}}$).

6. Conclusión sobre el método de los diagramas de Karnaugh

El método de los diagramas de Karnaugh que hemos expuesto permite obtener de forma elegante y rápida la simplificación de los circuitos lógicos de tres, cuatro, cinco y seis variables.

Por encima de seis variables, el método tiene una difícil utilización debido al gran número de diagramas que hay que manipular. Para estos problemas, se utilizará el método de Quine y de McCluskey, que no es otra cosa que un método general, no gráfico, de los diagramas de Karnaugh.

Este método es naturalmente válido para las funciones de tres y cuatro variables. Para las funciones de cinco o seis variables, o más, los cálculos son largos, pero realizables. Pueden efectuarse en ordenador.

7. El método de Quine y McCluskey

Este método se utilizará para la simplificación de los circuitos. Primeramente se determina la función lógica del circuito que se va a simplificar bajo la forma de una suma standard. A continuación se sustituyen los términos de la suma por sus equivalentes binarios. Después se agrupan los términos del mismo índice. Por índice se entiende el número de 1 contenidos en cada uno de los términos de la suma standard. Se establece a continuación comenzando por el término de índice cero (sólo hay un término de índice cero) los términos de índice 1, los términos de índice 2, etc., cada grupo está separado por un trazo horizontal. Cada término se referencia por su equivalente decimal.

A continuación se comparan los términos cuyo índice no difieren más que en 1, es decir los términos de índice cero con cada uno de los términos de índice 1, los términos de índice 1 con cada uno de los términos de índice 2 y así sucesivamente hasta finalizar la lista.

Combinamos entonces todos los términos susceptibles de combinarse. Sólo pueden combinarse los términos que sólo difieren en un dígito de acuerdo con la regla siguiente $AB + \bar{A}B = B$. Sólo se mantendrán los términos constantes, el término variable será eliminado sustituyéndolo por un guión. Se establece así una segunda lista de los términos que se referenciarán por el valor decimal de los dos términos de la primera lista donde se han originado.

Se señalarán todos los términos de la primera lista que se han combinado para dar nacimiento a la segunda lista. A continuación se verificará si pueden combinarse algunos términos de la segunda lista. En caso afirmativo se establece una

tercera lista procediendo de la misma manera que para el establecimiento de la segunda.

Se observará que las listas sacadas del orden se clasifican automáticamente por orden de índice creciente. La tercera lista contiene dos veces los mismos términos. Sólo se conservará uno, pues $A + A = A$. Si algunos términos de la tercera lista pueden combinarse todavía, se establecerá una cuarta lista. Cada término de esta lista se repetirá tres veces. Sólo se conservará uno, pues $A + A + A = A$.

Se procederá de esta forma de lista en lista hasta la obtención de una lista en la que no se podrá combinar ningún término.

La solución se representa entonces por la suma lógica de los términos que no han sido objeto de combinación.

A continuación se comprobará que esta suma es óptima. En efecto, la solución así obtenida no es obligatoriamente la más simple. Algunos términos de la suma pueden ser superfluos. Para determinar estos términos superfluos se utiliza una tabla de selección, que sólo conserva los términos *esenciales* de la suma.

El establecimiento y utilización de esta tabla se explica en los ejemplos siguientes:

Ejemplo 1. *Establecer mediante el método de Quine y de McCluskey un esquema equivalente, simplificado al de la figura 118.*

La funcion X a la salida tiene como valor:

$$X = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C ,$$

que se simplifica mediante el álgebra de Boole a:

$$X = \bar{A}\bar{C} + A\bar{B} .$$

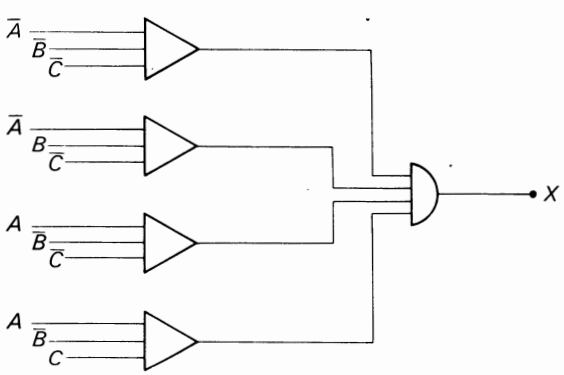


Fig. 118.

Está bajo la forma de una suma standard. Esto es lo que nos interesa. Sustituyamos cada uno de los términos de la suma por su equivalente binario:

$$X = \sum 000, 010, 100, 101 .$$

Establecemos una lista de términos por orden creciente de índice que se señalará por sus equivalentes decimales.

	Primera lista	Segunda lista	
0	0 0 0	0, 2	0 — 0
2	0 1 0	0, 4	—0 0
4	1 0 0	4, 5	1 0—
5	1 0 1		

Combinemos el término de índice cero con todos los términos de índice 1: los términos señalados por 0 y 2 se combinan para dar 0 — 0. El guión indica la ausencia de la variable *B*. Este término está señalado por 0, 2 en la segunda lista. Igualmente los términos señalados por 0 y 4 en la primera lista se combinan para dar — 00 en la segunda lista. Este término se señala por 0, 4. Así se ha combinado el término de índice cero con todos los términos de índice 1 con los de índice 2.

Los términos señalados por 2 y 5, en la primera lista, no pueden combinarse. *Difieren entre sí en más de dos dígitos.* Por el contrario, los términos 4 — 5 se combinan para dar 10 — en la segunda lista. Este último término se señala por 4, 5. Se han marcado con $\surd$ todos los términos que han entrado en una combinación en la primera lista.

En la segunda lista sólo se pueden combinar términos cuyas funciones están alineadas y sólo difieren en un dígito o una variable entre sí. Se observará que algún término de la segunda lista no puede combinarse. El resultado simplificado de *X* es la suma de los términos de la segunda lista, sea:

$$X = \bar{A}\bar{C} + \bar{B}\bar{C} + A\bar{B}\,.$$

Hay que verificar ahora que esta suma es la más simple, *en particular que ningún término de esta suma recubre los otros.* Para obtener este resultado, se establece lo que se llama *tabla de selección.*

Se dispondrán horizontalmente los equivalentes decimales que han asignado los términos de la segunda lista en el orden creciente y verticalmente los términos de esta lista (Fig. 119).

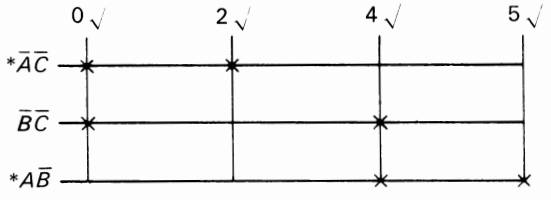


Fig. 119. *Tabla de selección.*

El término $\bar{A}\bar{C}$ se señala por 0 y 2 en la segunda lista. Se introduce una cruz en la intersección de las rectas $\bar{A}\bar{C}$, 0 y 2. Igualmente $\bar{B}\bar{C}$ se señala por una cruz en la intersección de las rectas $\bar{B}\bar{C}$, 0 y 4. Finalmente $A\bar{B}$ se señala por una cruz en la intersección de las rectas $A\bar{B}$, 4 y 5. Bajo esta forma la tabla se llama "tabla de selección". Se procederá entonces de la siguiente forma: al principio se buscan los *términos esenciales* de la suma. Por *términos esenciales se entienden los términos que dan nacimiento a una sola cruz en una columna.* Procediendo de izquierda a derecha en la figura 119 se observará que sólo existe una cruz en la

columna 2. Esta cruz se relaciona a $\overline{A}\overline{C}$. Este término es esencial. Se le atribuirá un asterisco $*\overline{A}\overline{C}$. Se asignarán todas las columnas relativas a $*\overline{A}\overline{C}$, sea 0 y 2 en nuestro caso. La segunda columna conteniendo sólo una cruz es 5. Esta columna se relaciona al término $A\overline{B}$ de la suma. $A\overline{B}$ es entonces el segundo término esencial de la suma. Se le designa por $*A\overline{B}$. Se cruzan las columnas 4 y 5 relativas a $*A\overline{B}$.

Todas las columnas se cubren por los términos esenciales $*\overline{A}\overline{C}$ y $*A\overline{B}$; $\overline{B}\overline{C}$ *es redundante.* Se engloba por un lado por $\overline{A}\overline{C}$ y por otro por $A\overline{B}$. La suma más simplificada que se puede obtener es:

$$X = \overline{A}\overline{C} + A\overline{B}$$

que corresponde al circuito de la figura 120:

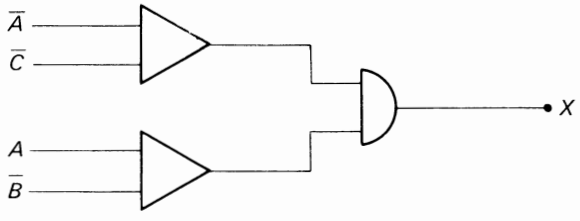


Fig. 120. *Esquema equivalente al de la figura* 118.

Se halla así el mismo resultado que el que se obtendrá mediante el álgebra de Boole o los diagramas de Karnaugh. En este último caso el diagrama de Karnaugh será el de la figura 121:

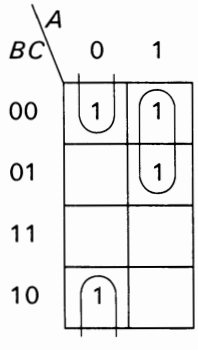


Fig. 121. $X = \overline{A}\overline{C} + A\overline{B}$.

Ejemplo 2. *Sea un circuito correspondiente a la función "booleana".*

$$X = \sum 0, 1, 2, 4, 5, 6, 9, 13 .$$

Simplificar este circuito mediante el método de Quine y McCluskey.

Bajo la forma binaria la expresión X tiene como valor:

$$X = \sum 0000, 0001, 0010, 0100, 0101, 0110, 1001, 1101 .$$

Estableceremos la primera lista en función del orden creciente de los índices. Combinemos a continuación los términos cuyo índice sólo difiere en una unidad. Se obtiene entonces la segunda lista:

Primera lista		Segunda lista	
0	0 0 0 0 √	0, 1	0 0 0 — √
1	0 0 0 1 √	0, 2	0 0 — 0 √
2	0 0 1 0 √	0, 4	0 — 0 0 √
4	0 1 0 0 √	1, 5	0 — 0 1 √
5	0 1 0 1 √	1, 9	— 0 0 1 √
6	0 1 1 0 √	2, 6	0 — 1 0 √
9	1 0 0 1 √	4, 5	0 1 0 — √
13	1 1 0 1 √	4, 6	0 1 — 0 √
		5, 13	— 1 0 1 √
		9, 13	1 — 0 1 √

Esta segunda lista contiene los términos de índice cero, 1 y 2 que pueden combinarse. Para que puedan combinarse es necesario que posean las mismas variables, *es decir hace falta que las funciones se alineen verticalmente.* Finalmente para las otras variables, éstas no pueden diferir más que en el valor de un dígito. Se establece así una tercera lista.

Se escogerán todos los términos de la segunda lista que pueden combinarse.

Tercera lista	
0, 1, 4, 5	0 — 0 —
0, 2, 4, 6	0 — — 0
~~0, 1, 4, 5~~	~~0 — 0 —~~
~~0, 2, 4, 6~~	~~0 — — 0~~
1, 5, 9, 13	— — 0 1
~~1, 9, 5, 13~~	~~— 0 1~~

La tercera lista contiene siempre los términos idénticos. Se suprimirá uno. El resultado será la suma de los términos que no han sido objeto de combinación, es decir los términos no escogidos en las tres listas. Obtenemos:

$$X = \bar{A}\bar{C} + \bar{A}\bar{D} + \bar{C}D\,.$$

Verifiquemos si esta suma es la más simple mediante la tabla de selección. Se observará que sólo los términos $\bar{A}\bar{D}$ y $\bar{C}D$ son esenciales. El término $\bar{A}\bar{C}$ es superfluo (Fig. 122).

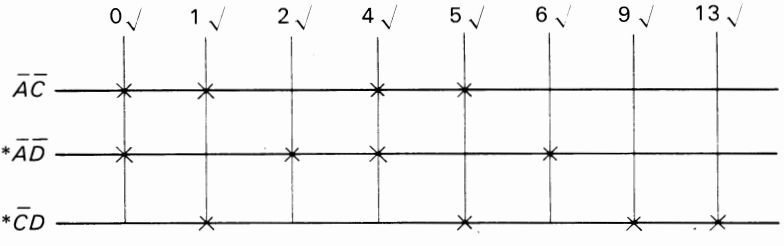


Fig. 122. *Tabla de selección para* $X = \bar{A}\bar{C} + \bar{A}\bar{D} + \bar{C}D$

La solución más simple es:

$$X = \overline{A}\overline{D} + \overline{C}D\,,$$

que corresponde al esquema equivalente de la figura 123. El diagrama de Karnaugh es el de la figura 124.

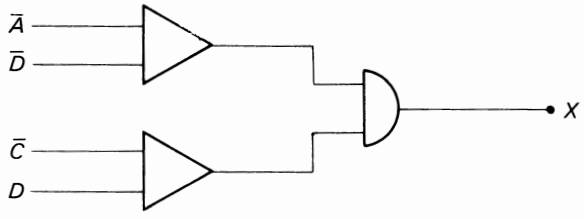


Fig. 123. *Esquema equivalente para* $X = \sum 0, 1, 2, 4, 5, 6, 9, 13.$

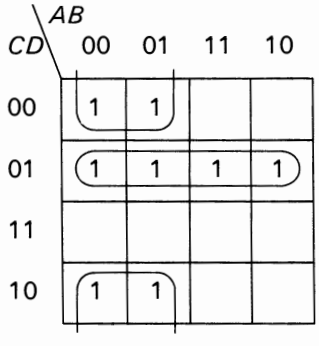


Fig. 124. $X = \overline{A}\overline{D} + \overline{C}D.$

Ejemplo 3. *Realizar el circuito más simple que responda a la función lógica:* $X = \sum 1, 5, 8, 9, 10, 11, 13, 15.$

Se utilizará el método de Quine y McCluskey.

La función a realizar bajo forma binaria es:

$$X = \sum 0001,\ 0101,\ 1000,\ 1001,\ 1010,\ 1011,\ 1101,\ 1111\ .$$

Establecemos la primera lista en función del orden creciente de los índices de cada uno de los términos y combinemos los términos posibles. Se obtiene así la segunda lista.

Esta da origen a una tercera lista. Después de la eliminación de los términos duplicados de la tercera lista, se obtiene:

Primera lista		Segunda lista	
1	0 0 0 1 √	1, 5	0 — 0 1 √
8	1 0 0 0 √	1, 9	— 0 0 1 √
5	0 1 0 1 √	8, 9	1 0 0 — √
9	1 0 0 1 √	8, 10	1 0 — 0 √
10	1 0 1 0 √	5, 13	— 1 0 1 √
11	1 0 1 1 √	9, 11	1 0 — 1 √
13	1 1 0 1 √	9, 13	1 — 0 1 √
15	1 1 1 1 √	10, 11	1 0 1 — √
		11, 15	1 — 1 1 √
		13, 15	1 1 — 1 √

Tercera lista	
8, 9, 10, 11	1 0 — —
~~8, 9, 10, 11~~	~~1 0 — —~~
1, 5, 9, 13	— — 0 1
~~1, 5, 9, 13~~	~~— — 0 1~~
9, 11, 13, 15	1 — — 1
~~9, 15, 11, 15~~	~~1 — — 1~~

$$X = A\overline{B} + \overline{C}D + AD\ .$$

Determinemos ahora la tabla de selección (Fig. 125).

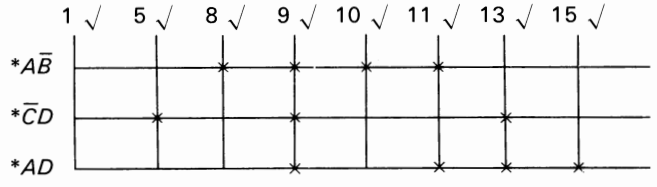


Fig. 125. *Tabla de selección para* $X = A\overline{B} + \overline{C}D + AD$.

Los tres términos de X son términos esenciales. No pueden eliminarse ninguno. El esquema equivalente es el de la figura 126.

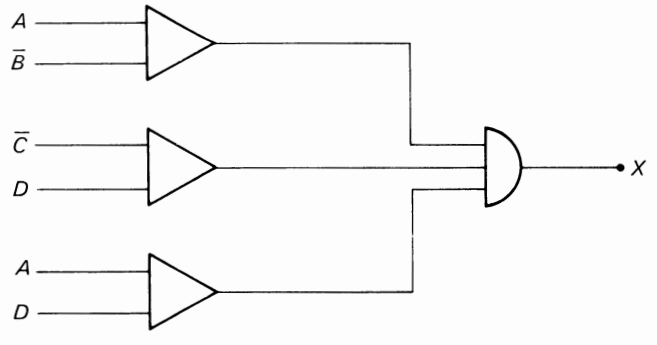


Fig. 126. *Esquema equivalente para* $X = \sum 1, 5, 8, 9, 10, 11, 13, 15$.

El diagrama de Karnaugh es el de la figura 127.

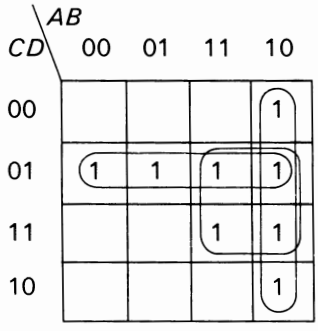


Fig. 127. $X = A\overline{B} + AD + \overline{C}D$.

Caso en que existan condiciones prohibidas

En este caso se atribuye a estas condiciones el valor arbitrario 0 = 1, después se utilizará el método de la misma manera que anteriormente. A continuación se señalarán en la tabla de selección, una vez simplificadas las condiciones prohibidas, si las hay: se les atribuirá entonces el valor cero. Se obtiene así la solución más simple correspondiente al diagrama de Karnaugh.

Ejemplo 4. *Realizar el circuito más simple que corresponda a la función lógica:*

$$X(A, B, C, D) = \sum 5,7 .$$

Las condiciones: 8*, 13*, *y* 15* *están prohibidas. Se las designa mediante un asterisco;*

Introduciendo las condiciones prohibidas en X obtenemos:

$$X = \sum 5, 7, 8^*, 13^*, 15^*$$

de donde se obtiene la primera lista:

8*	1 0 0 0	
5	0 1 0 1	√
7	0 1 1 1	√
13*	1 1 0 1	√
15*	1 1 1 1	√

y la segunda lista:

5, 7	0 1 — 1	√
5, 13*	— 1 0 1	√
7, 15*	0 1 1 1	√
13*, 15*	1 1 — 1	√

Combinando los términos de la segunda lista obtenemos una tercera lista:

5, 7, 13*, 15* — 1 — 1

~~5, 13*, 7, 15* — 1 — 1~~

El resultado final es la suma de los términos no borrados.

$$X = A\bar{B}\bar{C}\bar{D} + BD\ .$$

Se observará que el término $A\bar{B}\bar{C}\bar{D}$ (1000) es una condición prohibida. Atribuyéndole el valor 0 obtenemos el valor más simple para X:

$$X = BD\ .$$

En resumen, en el caso de condiciones prohibidas se incorporan totalmente a la suma ΣP_i. Se simplifica por el método de comparación de listas. Se escribe a continuación X y se comprueba que existen todavía en este resultado condiciones prohibidas. Estas corresponden a las que no han sido englobadas en los bucles de combinaciones, es decir a aquellas que tienen el valor cero. El resultado simplificado de X, se introduce a continuación en la tabla de selección y se procede como anteriormente.

El diagrama de Karnaugh correspondiente a la función estudiada es el de la figura 128.

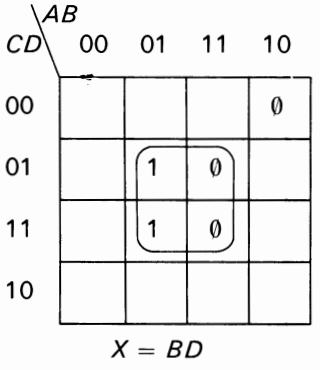


Fig. 128. *Diagrama de Karnaugh del problema* 4.

Ejemplo 5. *Realizar el circuito más simple correspondiente a la función lógica:*

$$X = \sum 2, 3, 4, 5\ .$$

Las siguientes condiciones son prohibidas: 10*, 11*, 12*, 13*, 14*, 15*.

Introducimos las condiciones prohibidas en la suma ΣP_i:

$$X = \sum 2, 3, 4, 5, 10^*, 11^*, 12^*, 13^*, 14^*, 15^*$$

y después de haber sustituido los números decimales por sus equivalentes binarios y separados éstos según sus índices, obtenemos la primera lista:

2	0 0 1 0 √
4	0 1 0 0 √
3	0 0 1 1 √
5	0 1 0 1 √
10*	1 0 1 0 √
12*	1 1 0 0 √
11*	1 0 1 1 √
13*	1 1 0 1 √
14*	1 1 1 0 √
15*	1 1 1 1 √

Combinando los términos cuyos índices sólo difieren en una unidad, obtenemos la segunda lista, después la tercera lista:

2, 3	0 0 1 — √
2, 10*	— 0 1 0 √
4, 5	0 1 0 — √
4, 12*	— 1 0 0 √
3, 11*	— 0 1 1 √
5, 13	— 1 0 1 √
10*, 11*	1 0 1 — √
10*, 14*	1 — 1 0 √
12*, 13*	1 1 0 — √
12*, 14*	1 1 — 0 √
11*, 15*	1 — 1 1 √
13*, 15*	1 1 — 1 √
14*, 15*	1 1 1 — √

Segunda lista

4, 5, 12*, 13*	— 1 0 —
~~4, 12*, 5, 13*~~	~~— 1 0 —~~
2, 10*, 3, 11*	— 0 1 —
~~2, 3, 10*, 11*~~	~~— 0 1 —~~
10*, 11*, 14*, 15*	1 — 1 —
~~10*, 14*, 11*, 15*~~	~~1 — 1 —~~
12*, 13*, 14*, 15*	1 1 — —
~~12*, 14*, 13*, 15*~~	~~1 1 — —~~

Tercera lista

No hay cuarta lista. Se observará que la tercera lista tiene combinaciones que sólo comprenden condiciones prohibidas. Esto equivale, en el diagrama de Karnaugh, a bucles que únicamente comprenden condiciones prohibidas: estos bucles son superfluos. Sólo las condiciones prohibidas 12* y 13* así como 10* y 11* están comprendidas en los bucles separados. Se atribuirá entonces el valor 0 a las condiciones 14* y 15*. Para solución de X queda:

$$X = B\overline{C} + \overline{B}C = B \oplus C \, .$$

No es necesario, en este caso, recurrir a la tabla de selección. El diagrama de Karnaugh se da en la figura 129.

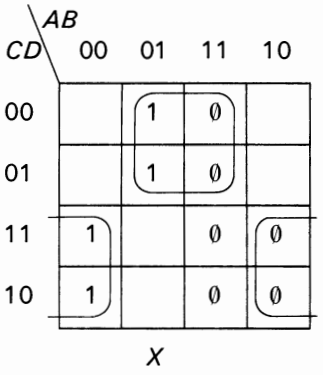


Fig. 129. *Diagrama de Karnaugh del ejemplo* 5 ($X = B \oplus C$).

Ejemplo 6. *Realizar el circuito más simple correspondiente a la función lógica:*

$$X = \sum 2, 4, 5, 8, 12, 13 ,$$

con las siguientes condiciones prohibidas: 0*, 7*, 10*, 15*.

Introduzcamos las condiciones prohibidas en X:

$$X = \sum 0^*, 2, 4, 5, 7^*, 8, 10^*, 12, 13, 15^* .$$

Establezcamos la primera y segunda lista:

0*	0 0 0 0	√
2	0 0 1 0	√
4	0 1 0 0	√
8	1 0 0 0	√
5	0 1 0 1	√
10*	1 0 1 0	√
12	1 1 0 0	√
7*	0 1 1 1	√
13	1 1 0 1	√
15*	1 1 1 1	√

Primera lista

0*, 2	0 0 — 0	√
0*, 4	0 — 0 0	√
0*, 8	— 0 0 0	√
2, 10*	— 0 1 0	√
4, 5	0 1 0 —	√
4, 12	— 1 0 0	√
8, 10*	1 0 — 0	√
8, 12	1 — 0 0	√
5, 7*	0 1 — 1	√
5, 13	— 1 0 1	√
12, 13	1 1 0 —	√
7*, 15*	— 1 1 1	√
13, 15*	1 1 — 1	√

Segunda lista

Finalmente, la tercera lista:

0*, 2, 8, 10*	— 0 — 0
0*, 4, 8, 12	— — 0 0
~~0*, 8, 2, 10*~~	~~— 0 — 0~~
~~0*, 8, 4, 12~~	~~— 0 0 —~~
4, 5, 12, 13	— 1 0 —
~~4, 12, 5, 13~~	~~— 1 0 —~~
5, 7*, 13, 15*	— 1 — 1
~~5, 13, 7*, 15*~~	~~— 1 — 1~~

No hay cuarta lista. En estas condiciones obtenemos:

$$X = \bar{B}\bar{D} + \bar{C}\bar{D} + B\bar{C} + BD\,.$$

Eliminaremos los términos redundantes, si los hay, mediante la tabla de selección. Hagamos las siguientes observaciones:

- $\bar{B}\bar{D}$ es un término esencial, pero no engloba las condiciones 12 y 13;
- para cumplir las condiciones 12 y 13 hay que mantener $B\bar{C}$;
- conservando $\bar{B}\bar{D}$ y $B\bar{C}$ el término $\bar{C}\bar{D}$ es redundante: en efecto, todas las combinaciones de este término están en $\bar{B}\bar{D}$ y $B\bar{C}$;
- las únicas combinaciones que no se han utilizado son 7* y 15* comprendidas en BD. Atribuyéndoles el valor cero, el término BD se convierte en $B\bar{C}D$ que es redundante y superfluo. Obtenemos finalmente:

$$X = \bar{B}\bar{D} + B\bar{C}\,.$$

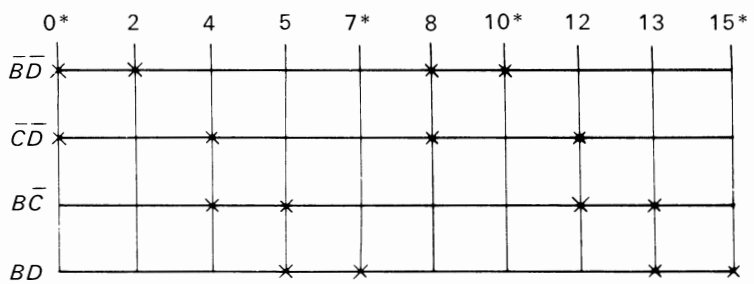


Fig. 130. *Tabla de selección del problema* 6.

El diagrama de Karnaugh se da en la figura 131:

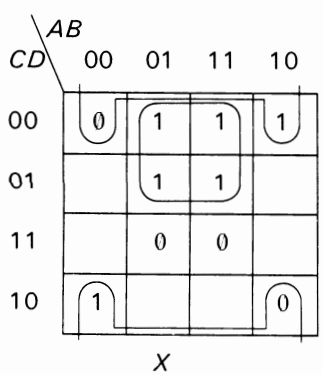


Fig. 131. *Diagrama de Karnaugh del problema* 6 ($X = \bar{B}\bar{D} + B\bar{C}$).

Capítulo 3

Aritmética binaria

1. Generalidades

Estamos acostumbrados a razonar durante muchos años en el sistema decimal. Este sistema tiene grandes ventajas: representación de grandes números con pocos símbolos, facilidad en los cálculos. Desgraciadamente un ordenador no puede utilizar el simbolismo decimal. Los circuitos electrónicos utilizados en los ordenadores sólo permiten distinguir dos estados.

Las primeras máquinas utilizaban relés mecánicos. Un relé sólo puede estar abierto o cerrado. Más tarde se han sustituido los relés por válvulas y transistores que, igualmente, sólo pueden tomar dos estados diferentes: una válvula o un transistor están o saturados o bloqueados. En el primer caso se comportan como un circuito cerrado y en el segundo como un circuito abierto. Realizan el papel de interruptores. La aritmética y los cálculos que realiza el ordenador estarán condicionados por esta naturaleza particular de los circuitos. Estos sólo reconocen o toman dos estados a los que se les atribuyen los valores 0 y 1 y que corresponden a un circuito abierto o cerrado. El sistema utilizado sólo tendrá dos símbolos, 0 y 1: éste es el sistema binario.

Los datos e instrucciones introducidas en la máquina estarán representados por una sucesión de 0 y 1. *La base del sistema de numeración que se utiliza es el número de símbolos de que dispone el utilizador del sistema.* Así, en el sistema decimal se dispone de diez símbolos 0, 1, 2, 3, ..., 9, para representar un número. La base *del sistema es* 10. En el sistema binario sólo se dispone de dos símbolos 0 y 1: *la base del sistema binario es* 2.

2. Consideraciones sobre el sistema decimal

El sistema decimal que acabamos de ver tiene diez símbolos. Todos los números habrán de representarse con estos diez símbolos. No existen dificultades para representar los números comprendidos entre 0 y 9. Para los números superiores tendremos que utilizar los mismos símbolos. La única forma de representar números superiores a 9 en el sistema decimal *consiste en cambiar la posición relativa a los símbolos.*

Por encima del 9 se toman los diez símbolos precedidos de la primera cifra significativa, es decir, 1: 10, 11, 12, 13, 14, ..., 19. Por encima del 19 se toman los diez símbolos precedidos de la segunda cifra significativa, el 2: 20, 21, 22, 23, 24, ..., 29 y así sucesivamente hasta 99. Por encima del 99 se necesita disponer de tres cifras. Se repiten entonces las 100 combinaciones anteriores precedidas de la

primera cifra significativa 100, 101, 102, ..., 199 y así sucesivamente hasta 999. La figura 1 resume la notación decimal de los mil primeros números (0 al 999).

0	10	20....	100....	990
1	11	21	101	991
2	12	22	102	992
3	13	23	103	993
4	14	24	104	994
5	15	25	105	995
6	16	26	106	996
7	17	27	107	997
8	18	28	108	998
9	19	29	109	999

Fig. 1. *Forma de representar los* 1 000 *primeros números decimales.*

Se observará que cualquiera que sea la magnitud del número, éste se representa siempre únicamente por diez símbolos, los cuales cambian de posición para cada decena. Al cambiar de posición toman un valor diferente.

Así, los 1 de

$$\begin{matrix} 1 \\ 10 \end{matrix}$$

no tienen el mismo significado, el primero representa una unidad, el segundo una decena. Se dice que no poseen el mismo *peso. Se atribuye así a los nueve símbolos significativos del sistema decimal pesos que varían con su posición en el número. A la posición más a la derecha del número se le atribuye el peso* 1 *ó* 10^0 *a la posición inmediatamente a la izquierda el peso* 10^2, *a la cuarta posición el peso* 10^3, ..., *a la enésima posición el peso* 10^{n-1}.

El valor del símbolo es entonces el producto del símbolo por el peso afectado a su posición. El número es la suma de cada uno de los productos parciales. En la figura 2 hemos representado el número 1967. El símbolo 7 tiene el peso 1, el símbolo 6 el peso 10, el símbolo 9 el peso 100, el símbolo 1 el peso 1 000. El valor exacto de cada uno de los símbolos será:

$$\begin{aligned} 1 \times 10^3 &= 1\,000 \\ 9 \times 10^2 &= \quad 900 \\ 6 \times 10 &= \quad 60 \\ 7 \times 10^0 &= \quad 7 \\ \text{Total} &: \quad 1\,967 \end{aligned}$$

y el resultado final será así:

$$1\,967 = 1 \times 10^3 + 9 \times 10^2 + 6 \times 10^1 + 7 \times 10^0 .$$

10^3	10^2	10^1	10^0
1	9	6	7

Fig. 2. *Representación del número* 1967 *en el sistema decimal. Observad los pesos afectados a cada uno de los símbolos utilizados para representar este número.*

Todos los números decimales se expresan de la misma forma en función de las potencias de 10. En el caso de un número fraccionario se atribuye a los símbolos dispuestos a la derecha de la coma pesos inferiores a 1, es decir, 10^{-1}, 10^{-2}, 10^{-3}, ... Así, el número 3 592,750 se representará:

$$3\,592.750 = 3 \times 10^3 + 5 \times 10^2 + 9 \times 10^1 + 2 \times 10^0 + 7 \times 10^{-1} + 5 \times 10^{-2} ,$$

lo que corresponde a la tabla de la figura 3:

10^3	10^2	10^1	10^0	10^{-1}	10^{-2}
3	5	9	2	7	5

Fig. 3. *Representación de* 3592,75 *en el sistema decimal. Observad los pesos atribuidos a los diferentes símbolos del número.*

De una forma general un número decimal se representa por:

$$X_n\, X_{n-1}\, X_{n-2} \ldots X_0,\, X_{-1}\, X_{-2}\, X_{-3} \ldots X_m$$

se expresará así:

$$X_n\, X_{n-1}\, X_{n-2} \ldots X_0,\, X_{-1} \ldots X_{-m} = X_n\, 10^{n-1} + X_{n-1}\, 10^{n-2} \ldots X_0 + \\ + X_{-1}\, 10^{-1} + X_{-2}\, 10^{-2} + \cdots + X_{-m}\, 10^{-m} .$$

3. El sistema binario

En este sistema sólo se dispone de *dos* símbolos 1 y 0 para representar un número. Este se representará siempre por una sucesión de 1 y 0. Se utiliza como

en el sistema decimal, el desplazamiento hacia la izquierda del único símbolo significativo cuando se desea contar más allá de 1, 3, 7, 15, 31, etc. El cuadro de la figura 4 da las equivalencias binarias de los veinte primeros números decimales. Se observará que para contar por encima de 1 tenemos dos símbolos 10, 11 que son los equivalentes binarios de los mismos decimales 2 y 3. Para contar más allá de 3, se cuenta con las cuatro combinaciones precedentes a lo que se les añade a la izquierda el único símbolo significativo, lo que da 100, 101, 110, 111, es decir los equivalentes binarios de los números decimales 4, 5, 6, 7. Se llama *dígito* a cada uno de los símbolos 0 ó 1 utilizados para representar un número en forma binaria. Así el equivalente binario de 5 = 101 comprende 3 dígitos.

Decimal	Binario	Decimal	Binario
0	0	11	1 0 1 1
1	1	12	1 1 0 0
2	1 0	13	1 1 0 1
3	1 1	14	1 1 1 0
4	1 0 0	15	1 1 1 1
5	1 0 1	16	1 0 0 0 0
6	1 1 0	17	1 0 0 0 1
7	1 1 1	18	1 0 0 1 0
8	1 0 0 0	19	1 0 0 1 1
9	1 0 0 1	20	1 0 1 0 0
10	1 0 1 0		

Fig. 4. *Equivalentes binarios de los veinte primeros números decimales.*

Siendo 2 la base del sistema binario, a cada dígito se le afectan los pesos correspondientes 2^0, 2^1, 2^2, 2^3, ..., el peso 2^0 corresponde al dígito más a la derecha. Así, nosotros tendremos:

$$101 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 5\,,$$

$$1001 = 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 9\,.$$

En el caso de números fraccionarios, se coloca un punto entre la parte entera del número binario y su parte fraccionaria. Se atribuye entonces a los dígitos de la parte fraccionaria los pesos más pequeños que 1, como 2^{-1}, 2^{-2}, 2^{-3}, ... Así, tendremos por ejemplo:

$$11.011 = 1 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = 3\ \ 3/8$$

$$11.011 = 3\ \ 3/8\,.$$

o:

$$0.101 = 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = 5/8 .$$

De una forma general, si un número binario se representa bajo la forma de una sucesión de dígitos, tendremos:

$$X_n X_{n-1} \ldots X_0, X_{-1} X_{-2} \ldots X_{-m} = X_n 2^{n-1} + X_{n-1} 2^{n-2} \ldots X_0, X_{-1} 2^{-1} X_{-2} 2^{-2} \ldots X_{-m} 2^{-m} .$$

3.1 Adición y substracción binarias

La adición y la substracción binaria se efectuarán de la misma forma que la adición y substracción decimales. Es suficiente recordar la tabla de la adición y la substracción que son muy simples pues sólo tienen dos símbolos (Fig. 5):

$0 + 0 = 0$	$0 - 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$
$1 + 0 = 1$	$1 - 1 = 0$
$1 + 1 = 0$ retiene 1	$0 - 1 = 1$ retiene 1 .

Fig. 5. *Tablas de adición y substracción binaria.*

Ejemplo 1. *Sea sumar* 5 *y* 6 *en el sistema binario.* Obtenemos:

$$\begin{array}{r} 1\,0\,1 \\ +\ 1\,1\,0 \\ \hline 1\,0\,1\,1 \end{array}$$

El resultado 1011 equivale en binario al 11 del sistema decimal. Se observará la existencia de un resto que nace en el rango de peso más elevado y que pasa al orden superior.

Ejemplo 2. *Sea restar* 3 *de* 16. Tenemos:

$$\begin{array}{r} 1\,0\,0\,0\,0 \\ -\ \ \ \ 1\,1 \\ \hline 0\,1\,1\,0\,1 \end{array}$$

Se observará en este ejemplo la existencia de un resto en el orden más pequeño que se desplaza seguidamente a todos los órdenes. La operación se termina cuando el resto se ha trasladado de orden en orden hasta el último. Encontraremos este problema en el estudio de los circuitos de adición y substracción.

3.2 Multiplicación y división binaria

Las tablas de multiplicación y división binaria son inmediatas (Fig. 6):

$$\begin{aligned} 0 \times 0 &= 0 \qquad 0 : 1 = 0 \\ 0 \times 1 &= 0 \qquad 1 : 1 = 1 \\ 1 \times 0 &= 0 \\ 1 \times 1 &= 1 \end{aligned}$$

Fig. 6. *Tabla de multiplicación y división binaria.*

Se observará que la multiplicación de dos dígitos, uno de los cuales es cero, da siempre 0.

Ejemplo 1. *Sea multiplicar* 10 *por* 5. Obtenemos:

$$\begin{array}{r} 1\,0\,1\,0 \\ \times\ \ 1\,0\,1 \\ \hline 1\,0\,1\,0 \\ 0\,0\,0\,0 \\ 1\,0\,1\,0 \\ \hline 1\,1\,0\,0\,1\,0 \end{array}$$

El resultado 110010 corresponde al 50 decimal. En la multiplicación binaria como en la multiplicación decimal, es posible eliminar todas las multiplicaciones por cero desplazando un orden el siguiente producto parcial.

Ejemplo 2. *Sea dividir* 10 *entre* 5. Obtenemos:

$$\begin{array}{r|l} 1\,0\,1\,0 & 1\,0\,1 \\ \cline{2-2} 1\,0\,1 & \ \ 1\,0 \\ \cline{1-1} 0\,0\,0\,0 & \end{array}$$

El resultado es 10 que corresponde a 2 en el sistema decimal.

Ejemplo 3. *Dividir* 9 *entre* 4. Obtenemos en este caso.

$$\begin{array}{r|l} 1\,0\,0\,1 & 1\,0\,0 \\ \cline{2-2} 1\,0\,0 & 10.01 \\ \cline{1-1} 0\,0\,0\,1\,0\,0 & \\ \quad\ \ 1\,0\,0 & \\ \cline{1-1} \quad\ \ 0\,0\,0 & \end{array}$$

El resultado es 10.01 que es equivalente al binario 2,25, que representa el resultado buscado.

Ejemplo 4. *Sea dividir* 5,625 *entre* 2,25, *es decir en binario dividir* 101.101 *entre* 10.01. Multiplicando estos dos números por 2^3 obtenemos 101101 y 10010 que dividimos. Esto equivale a suprimir la parte decimal en los dos números.

```
1 0 1 1 0 1  | 1 0 0 1 0
1 0 0 1 0    |   10.1
-------------|
0 0 1 0 0 1 0|
    1 0 0 1 0|
    ---------|
    1 0 0 0 0|
```

El resultado es 10.1, es decir 2,5 que representa el valor decimal del resultado.

4. Conversión decimal binario

En una calculadora los números que se tratan son binarios. Se necesita poder convertir un número decimal sea grande o pequeño.

4.1 Primer método, válido para números pequeños

Hemos visto anteriormente que un número decimal se obtiene haciendo la suma de los productos parciales de las potencias de dos por cada uno de los dígitos del número binario correspondiente.

Inversamente, para calcular un número binario, a partir de un número decimal, se resta sucesivamente de este número las más altas potencias de dos, se atribuye el dígito 1 a estas potencias si existen y el dígito 0 si no existen.

Ejemplo. *Sea pasar* 25 *a binario.*

La más alta potencia de 2 que puede restarse de 25 es $2^4 = 16$. Se asociará el dígito 1 al peso 2^4 del número binario. Queda $25 - 16 = 9$. La mayor potencia de 2 que se puede restar es 2^3 del número binario. Queda entonces $(9 - 8 = 1$.

La mayor potencia de 2 que puede restarse del número 1 es $2^0 = 1$. Se le atribuye el dígito 1 al peso 2^0 del número binario. Se observará que no existen los pesos 2^2 y 2^1. Se les atribuye el dígito 0. El resultado es entonces el siguiente:

$$\begin{aligned} 25 - 2^4 &= 9 \\ 9 - 2^3 &= 1 \\ 1 - 2^0 &= 0 \end{aligned} \quad \Rightarrow \quad \begin{array}{c} 2^4\ 2^3\ 2^2\ 2^1\ 2^0 \\ \hline 1\ 1\ 0\ 0\ 1 \end{array}$$

Se observará que si n es la mayor potencia de 2 del número binario (aquí 4) el número binario tiene $n + 1$ términos.

4.2. Segundo método, válido para grandes números

Este método es el más práctico y se utilizará normalmente para convertir grandes números decimales. Consiste en dividir el número decimal que se va a convertir sucesivamente por 2.

La sucesión de los restos da el número binario que se busca: el último resto corresponde al peso más elevado; el primer resto al peso más débil.

Ejemplo. *Pasar a binario el número* 125:

125 : 2 = 62 +	Resta 1	
62 : 2 = 31	0 ↑	
31 : 2 = 15	1	Resultado
15 : 2 = 7	1	1 1 1 1 1 0 1
7 : 2 = 3	1	
3 : 2 = 1	1	
1 : 2 = 0	1	

Se comprobará que este resultado es exacto pasando el número binario a decimal.

Comprobación

$$\underbrace{1\,1\,1\,1\,1\,0\,1}_{7\text{ términos}} = (1\times 2^6)+(1\times 2^5)+(1\times 2^4)+(1\times 2^3)+(1\times 2^2)+(0\times 2^1)+(1\times 2^0)=125\,.$$

4.2.1 *Conversión de un número decimal fraccionario*

El método anterior no se aplica a los números decimales fraccionarios.

En este caso, se descompone el número en un número entero y un número fraccionario, calculándose por separado su representación binaria. Las dos partes binarias se colocan a continuación juntas. Necesitamos saber convertir un número fraccionario en su equivalente binario.

4.2.1.1 *Caso de números fraccionarios de pequeña longitud*

Se restarán sucesivamente del número fraccionario las potencias negativas más pequeñas hasta obtener la precisión deseada.

Ejemplo. *Pasar a binario el valor decimal* 0.875:

$$0.875 - (1 \times 2^{-1}) = 0.875 - 0.5 = 0.375$$
$$0.375 - (1 \times 2^{-2}) = 0.375 - 0.25 = 0.125$$
$$0.125 - (1 \times 2^{-3}) = 0.125 - 0.125 = 0\,.$$

El equivalente binario de 0.875 es 0.111.

4.2.1.2 *Caso de fracciones de gran longitud*

En este caso se multiplica por 2 la fracción decimal. Si aparece un 1 *a la izquierda del punto decimal* después de la multiplicación por 2 se añade un 1 a la fracción binaria en formación. Si después se multiplica por 2 aparece un 0 a la izquierda del número decimal, se añade un cero a la derecha del número binario.

Ejemplo. *Pasar a binario el número decimal* 0.4375.

	Representación binaria
$2 \times 0.4375 = 0.8750$	0.0
$2 \times 0.875 = 1.750$	0.0 1
$2 \times 0.750 = 1.50$	0.0 1 1
$2 \times 0.5 = 1.00$	0.0 1 1 1

La representación binaria de 0.4375 es 0.0111.

Ejemplo. *Pasar a binario el número decimal* 102,247.

Inicialmente dividimos 102.247 en dos partes.

$$\begin{matrix} 102 \\ 0.247 \end{matrix}$$

calculándose por separado sus equivalentes binarios.

Para la parte entera:

$$\begin{array}{rcl} 102 : 2 = 51 & \longrightarrow & 0 \uparrow \\ 51 : 2 = 25 & \longrightarrow & 1 \\ 25 : 2 = 12 & \longrightarrow & 1 \\ 12 : 2 = 6 & \longrightarrow & 0 \\ 6 : 2 = 3 & \longrightarrow & 0 \\ 3 : 2 = 1 & \longrightarrow & 1 \\ 1 : 2 = 0 & \longrightarrow & 1 \end{array} \qquad \underbrace{1\,1\,0\,0\,1\,1\,0}_{102_{10}}$$

Para la parte fraccionaria:

0.247	0.
$2 \times 0.247 = 0.494$	0.0
$2 \times 0.494 = 0.988$	0.0 0
$2 \times 0.988 = 1.976$	0.0 0 1
$2 \times 0.976 = 1.952$	0.0 0 1 1
$2 \times 0.952 = 1.904$	0.0 0 1 1 1
$2 \times 0.904 = 1.808$	0.0 0 1 1 1 1

de donde, asociando los resultados con la precisión requerida se tiene:

$$(102.247_{10}) = 1\,1\;\;0\,0\;\;1\,1\,0.0\,0\;\;1\,1\,1\,1$$

5. *Naturaleza de los números tratados por los registros*

En general los números tratados son de coma fija. La posición de la coma puede disponerse al final del número, después del primer peso. En este caso los números tratados son enteros. La posición de la coma puede ponerse inmediata-

mente después del primer peso que indica el signo del número. En este caso los números tratados son fraccionarios.

Si se utilizaran números enteros en la multiplicación, hay desbordamiento hacia los pesos de orden más elevado. Estos se pierden en algunos casos. Si se utilizaran números fraccionarios el desbordamiento se efectúa detrás del peso más débil y los dígitos perdidos modifican muy poco el resultado.

Por esta razón algunas máquinas sólo tratan números binarios inferiores a 1.

6. Representación binaria de los números con signo

En aritmética binaria como en aritmética ordinaria hay que distinguir un número negativo de un número positivo. Hay que atribuir un signo a todo número binario: se dice que es un número con *signo*.

Hay tres formas de representación de los números con signo, según la forma en que se tratan los números negativos:

- representación en valor absoluto (módulo),
- representación de los números negativos por sus complementos a 2,
- representación de los números negativos por sus complementos a 1.

El ordenador sólo puede distinguir dos estados: 0 y 1. Hay que atribuir al signo del número 1 estos dos valores. Por convenio, se le atribuye al valor 0 el signo + y al 1 el –. El signo, como en el álgebra ordinaria, se colocará delante del número en la posición de los pesos más fuertes:

Ejemplo:

0 | 1 0 1 0 (+ 10)
1 | 1 0 1 0 (– 10)
0 | 0 1 0 0 (+ 4)
1 | 0 1 0 0 (– 4)

Esta representación se utiliza siempre en las máquinas. Las operaciones se realizarán con estos números como en el álgebra ordinaria: se efectuarán las operaciones con los módulos y los signos se tratarán aparte. Consideremos un número mayor que 1, que posea m dígitos fraccionarios, $n-1$ dígitos en la parte entera, el signo se afecta a la posición n, el número se representará de la siguiente forma:

$$2^n \mid 2^{n-1}\ 2^{n-2}\ 2^{n-3} \ldots 2^0\ 2^{-1}\ 2^{-2} \ldots 2^{-m}$$

$$X_n \mid X_{n-1}\ X_{n-2}\ X_{n-3} \ldots X_0\ X_{-1}\ X_{-2} \ldots X_{-m}\ .$$

El valor algebraico del número X se obtendrá mediante la siguiente fórmula:

$$X = (-1)\, X_n \sum_{k=-m}^{n-1} X_k\, 2^k \qquad (1)$$

con:

$$X_n = 0 \quad \text{para} \quad X > 0$$
$$X_n = 1 \quad \text{para} \quad X < 0 .$$

La expresión (2) representa el valor algebraico de un número binario: representado en valor absoluto y en signo, cuando es superior a 1. En el caso en que todos los números representados en la máquina son inferiores a 1, $n = 0$, $m = n$, tenemos la siguiente fórmula:

$$2^0 \mid 2^{-1}\ 2^{-2} \ldots 2^{-n}$$
$$X_0 \mid X_{-1}\ X_{-2} \ldots X_{-n}$$

el valor algebraico del número se representará entonces por:

$$X = (-1)\, X_0 \sum_{k=1}^{n} X_{-k}\, 2^{-k} \qquad (2)$$

con:

$$X_0 = 0 \quad \text{para} \quad X > 0$$
$$X_0 = 1 \quad \text{para} \quad X < 0 .$$

6.1 Representación de los números negativos por sus complementos a 2

6.1.1 *Caso de números mayores que* 1

Representamos un número negativo por su complemento a 2, lo que, desde el punto de vista binario, equivale a cambiar los 0 en 1 y los 1 en 0 y sumar al peso más débil.

Ejemplo 1: 0 | 0 1 1 1 (+ 7)

El complemento a 2 se obtendrá de la siguiente forma:

```
  1 | 1 0 0 0
+             1
  ---------------
1 | 1 0 0 1  (- 7)
```

El signo 1 en el peso de más fuerza indica un número negativo.

Ejemplo 2:

```
  0 | 1 0 1 0  (+ 10)
  1 | 0 1 0 1
+             1
  ---------------
1 | 0 1 1 0  (- 10)
```

Observación importante. Si atribuimos valor (− 1) al dígito con signo de un número negativo representado por su complemento a 2 tendremos, para los ejem-

plos anteriores:

$$(-1) \times 2^4 + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) = -7,$$

$$(-1) \times 2^4 + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0) = -10.$$

Así, es suficiente atribuir al dígito del signo el valor – 1 para obtener el valor algebraico exacto del número.

Esta última propiedad se utiliza para representar los números negativos por su complemento a 2. Tendremos entonces:

$$X = 0 \times 2^n + \sum_{k=-m}^{n-1} X_k \, 2^k \qquad \text{para} \quad X > 0 \tag{3}$$

$$X = (-1) \times 2^n + \sum_{k=-m}^{n-1} \overline{X}_k \, 2^k + 2^{-m} \qquad \text{para} \quad X < 0 \tag{4}$$

con $$\overline{X}_k = 1 - X_k \,. \tag{5}$$

$\overline{X}_k$ representa el complemento a 1 de X_k. 2^{-m} representa la unidad añadida al peso más débil del número.

6.1.2 *Caso de números inferiores a* 1

Como anteriormente, tenemos entonces $n = 0$, $m = n$:

$$X = 0 \times 2^0 + \sum_{k=1}^{n} X_{-k} \, 2^{-k} \qquad \text{para} \quad X > 0 \tag{6}$$

$$X = (-1) \times 2^0 + \sum_{k=1}^{n} \overline{X}_{-k} \, 2^{-k} + 2^{-n} \qquad \text{para} \quad X < 0 \,. \tag{7}$$

donde $\overline{X}_k$ viene dado por 5.

6.2 Representación de los números negativos por sus complementos a 1

6.2.1 *Caso de números mayores que* 1

Tomar el complemento a 1 de un número equivale a cambiar los ceros y los 1.

Ejemplo:

0	|	0	1	1	1	(+ 7)
1	|	1	0	0	0	(– 7)
0	|	1	0	1	0	(+ 10)
1	|	0	1	0	1	(– 10)
1	|	1	1	1	1	(– 0)

La configuración 1 | 1 1 1 1 representa lo que se llama el cero negativo.

En este caso tenemos:

$$X = 0 \times 2^n + \sum_{k=-m}^{n-1} X_k 2^k \qquad \text{para } x > 0 \tag{8}$$

$$X = (-1) 2^n + \sum_{k=-m}^{n-1} \overline{X}_k 2^k \qquad \text{para } x < 0 . \tag{9}$$

6.2.2 *Caso de números inferiores a* 1

Tenemos, como anteriormente, $n = 0, m = n$. Los exponentes de los diferentes pares son negativos:

$$X = 0 \times 2^0 + \sum_{k=1}^{n} X_{-k} 2^{-k} \qquad \text{para} \quad X > 0 \tag{10}$$

$$X = (-1) 2^0 + \sum_{k=1}^{n} \overline{X}_{-k} 2^{-k} \qquad \text{para} \quad X < 0 . \tag{11}$$

6.2.3 *Relaciones interesantes*

Consideremos un número fraccionario mayor que 1 que tiene m dígitos fraccionarios y $(n-1)$ dígitos enteros todos iguales a 1. Añadamos 1 al peso más débil, obtenemos:

$$\begin{array}{cccccccccccc}
2^n & 2^{n-1} & \ldots\ldots\ldots & 2^0 & 2^{-1} & \ldots\ldots\ldots & 2^{-m} \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 & & & & & & & & & & + 1 \\
\hline
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

es decir:

$$2^n = \sum_{k=-m}^{n-1} 2^k + 2^{-m} . \tag{12}$$

En el caso de un número fraccionario inferior a la unidad $n = 0$, $m = n$, $k = -k$, de donde

$$2^0 = \sum_{k=1}^{n} 2^{-k} + 2^{-n} , \tag{13}$$

es decir:

$$\sum_{k=1}^{n} 2^{-k} = 1 - 2^{-n} . \tag{14}$$

7. Adición y substracción en el caso en que los números se presenten con módulo y signo

7.1 Caso de la adición

Se distinguirán dos casos según que los números sean de signos idénticos u opuestos. Si los dos signos son idénticos, la máquina hace la suma de los módulos y conserva los signos del número dispuesto en el acumulador, antes del proceso de adición. El primer número X está colocado en el acumulador. El acumulador es un dispositivo que permite sumar y restar dos números y retener el resultado.

El número a sumar Y se dispone en un registro memoria R, donde permanecerá durante la operación. Si los signos son opuestos, la máquina realiza la diferencia de los módulos. El resultado será positivo o negativo en función del valor de los módulos. El módulo del resultado será positivo o negativo en función del valor de los módulos. El módulo del resultado se obtendrá como complementario a 2 del módulo acumulador.

Naturalmente habrá que prever en el acumulador una posición de desbordamiento delante del signo que admita una suma total de orden más elevado que el módulo. Suponemos que el resultado no sobrepasa la capacidad del acumulador.

Los siguientes ejemplos muestran el método:

Ejemplo 1. *Caso de dos números positivos.*

	Sgn					
Acumulador (X)	0	|	1	0	1	0 (+ 10)
Registro de memoria (Y)	0	|	0	1	0	0 (+ 4)
Control de suma			1	0	1	0
Signos idénticos: Adición de módulos		+	0	1	0	0
			1	1	1	0
Sin suma superior al orden más elevado del módulo						
Conservación del signo del acumulador						
Resultado en el acumulador	0	|	1	1	1	0 (+ 14)

Ejemplo 2. *Caso de dos números negativos*

	Sgn					
Acumulador (X)	1	|	1	0	1	0 (– 10)
Registro de memoria (Y)	1	|	0	1	0	0 (– 4)
Control de suma			1	0	1	0
Signos idénticos: Adición de módulos		+	0	1	0	0
			1	1	1	0

Sin arrastre de suma superior al orden más elevado del módulo Conservación del signo del acumulador							
Resultado en el acumulador	1	|	1	1	1	0	(− 14)

Ejemplo 3. *Caso de dos números de signos opuestos,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

	Sgn						
Acumulador (X)	0	|	1	0	1	0	(+ 10)
Registro de memoria (Y)	1	|	0	1	0	0	(− 4)
Control de adición			1	0	1	0	
Signos diferentes: Substracción de módulos		−	0	1	0	0	
			0	1	1	0	
Sin suma anterior superior al orden más elevado del módulo Conservación del signo del acumulador							
Resultado en el acumulador	0	|	0	1	1	0	(+ 6)

Ejemplo 4. *Caso de dos números de signos contrarios,* $X > 0$, $Y < 0$ *con* $|X| < |Y|$.

	Sgn						
Acumulador (X)	0	|	0	1	0	0	(+ 4)
Registro de memoria (Y)	1	|	1	0	1	0	(− 10)
Control de adición		−	0	1	0	0	
Signos diferentes: Substracción de módulos			1	0	1	0	
Suma anterior superior al orden más elevado del módulo. Inversión del signo del acumulador. Módulo expresado por su complemento a 2.	C_0 ↷		1	0	1	0	
Resultado en el acumulador después de la inversión del signo.	1	|	1	0	1	0	
Complemento a 2 del módulo	1	|	0	1	1	0	(− 6)

Ejemplo 5. *Caso de dos números de signos contrarios,* $X < 0$, $Y > 0$, *con* $|X| > |Y|$.

	Sgn						
Acumulador (X)	1	|	1	0	1	0	(− 10)
Registro de memoria (Y)	0	|	0	1	0	0	(+ 4)
Control de adición		−	1	0	1	0	
Signos diferentes: Diferencia de módulos			0	1	0	0	
Sin suma anterior superior al orden más elevado del módulo. Conservación del signo del acumulador.			0	1	1	0	
Resultado en el acumulador	1	|	0	1	1	0	(− 6)

Ejemplo 6. *Caso de dos números de signos contrarios,* $X < 0$, $Y > 0$, *con* $|X| < |Y|$.

	Sgn						
Acumulador (X)	1	|	0	1	0	0	(− 4)
Registro de memoria (Y)	0	|	1	0	1	0	(+ 10)
Control de adición		−	0	1	0	0	
Signos diferentes: Diferencia de módulos			1	0	1	0	
Suma anterior superior al orden más elevado del módulo. Inversión del signo del acumulador. El módulo se expresa mediante el complemento a 2.	C_0 ↷		1	0	1	0	
Resultado en el acumulador después de la inversión del signo.	0	|	1	0	1	0	
Complemento a 2 del módulo	0	|	0	1	1	0	(+ 6)

7.2 Substracción

Existen ocho posibilidades según el signo y el módulo de los dos números. En el caso en que los dos números son de distinto signo se suman los módulos. El signo del resultado es el del número en el acumulador antes de la adición del módulo. Se conserva el signo del acumulador. Si los dos números son del mismo signo se efectúa la substracción de los módulos. Si no hay resultado superior al módulo no cambia el signo del módulo. En caso contrario se cambiará el signo y se tomará el complemento a 2 del módulo del acumulador.

Ejemplo 1. *Caso de dos números positivos,* $X > 0$, $Y > 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	0	0	1	0	0	(+ 4)
Control de substracción		–	1	0	1	0
Signos idénticos: Diferencia de módulos			0	1	0	0
Sin resto más allá del orden superior al módulo			0	1	1	0
Signo del acumulador conservado						
Resultado en el acumulador	0	0	1	1	0	(+ 6)

Ejemplo 2. *Caso de dos números positivos,* $X > 0$, $Y > 0$, *con* $|X| < |Y|$.

Acumulador (X)	0	0	1	0	0	(+ 4)
Registro de memoria (Y)	0	1	0	1	0	(+ 10)
Control de substracción		–	0	1	0	0
Signos idénticos: Diferencia de módulos			1	0	1	0
Con resto superior al módulo. Inversión del signo y complemento a 2 del módulo del acumulador	C_0 ↷		1	0	1	0
Resultado en el acumulador después de la inversión del signo	1	1	0	1	0	
Resultado en el acumulador después del complemento a 2 del módulo	1	0	1	1	0	(– 6)

Ejemplo 3. *Caso de dos números negativos,* $X < 0$, $Y < 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	1	1	0	1	0	(– 10)
Registro de memoria (Y)	1	0	1	0	0	(– 4)
Control de substracción		–	1	0	1	0
Signos idénticos: Diferencia de módulos			0	1	0	0
Sin resto. Conservación del signo del acumulador			0	1	1	0
Resultado en el acumulador	1	0	1	1	0	(– 6)

Ejemplo 4. *Caso de dos números negativos*, $X < 0$, $Y < 0$, *con* $|X| < |Y|$.

	Sgn						
Acumulador (X)	1	|	0	1	0	0	(− 4)
Registro de memoria (Y)	1	|	1	0	1	0	(− 10)
Control de substracción			0	1	0	0	
Signos idénticos: Diferencia de módulos		−	1	0	1	0	
Con resto superior al módulo. Inversión del signo y complemento a 2 del módulo del acumulador	C_0 ↶		1	0	1	0	
Resultado en el acumulador	0	|	1	0	1	0	
Resultado después del complemento del módulo	0	|	0	1	1	0	(+ 6)

Ejemplo 5. *Caso de dos números de signos contrarios* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

	Sgn						
Acumulador (X)	0	|	1	0	1	0	(+ 10)
Registro memoria (Y)	1	|	0	1	0	0	(− 4)
Control de substracción			1	0	1	0	
Signos diferentes: Suma de módulos		+	0	1	0	0	
Sin resto más allá del orden más elevado del módulo. Se conserva el signo del acumulador.			1	1	1	0	
Resultado en el acumulador	0	|	1	1	1	0	(+ 14)

Ejemplo 6. *Caso de dos números de signos opuestos*, $X > 0$, $Y < 0$, *con* $|X| < |Y|$.

	Sgn						
Acumulador (X)	0	|	0	1	.0	0	(+ 4)
Registro memoria (Y)	1	|	1	0	1	0	(− 10)
Control de substracción			0	1	0	0	
Signos diferentes: Suma de módulos		+	1	0	1	0	
Sin resto más allá del orden más elevado del módulo. Se conserva el signo del acumulador.			1	1	1	0	
Resultado en el acumulador	0	|	1	1	1	0	(+ 14)

Ejemplo 7. *Caso de dos números de signos distintos*, $X < 0$, $Y > 0$, *con* $|X| > |Y|$.

Acumulador (X)	1	|	1	0	1	0	(− 10)
Registro de memoria (Y)	0	|	0	1	0	0	(+ 4)
Control de substracción			1	0	1	0	
Signos diferentes: Suma de módulos		+	0	1	0	0	
Sin resto superior al orden más elevado del módulo. Se conserva el signo del acumulador			1	1	1	0	
Resultado en el acumulador							

Ejemplo 8. *Caso de dos números de signos contrarios*, $X < 0$, $Y > 0$, *con* $|X| < |Y|$.

Acumulador (X)	1	|	0	1	0	0	(− 4)
Registro de memoria (Y)	0	|	1	0	1	0	(+ 10)
Control de substracción			0	1	0	0	
Signos diferentes: Suma de módulos		+	1	0	1	0	
Sin resto superior al orden más elevado del módulo. Se conserva el signo del acumulador			1	1	1	0	
Resultado en el acumulador	1	|	1	1	1	0	**(− 14)**

7.2.1 *Recapitulación*

Los ejemplos anteriores dan todos los casos de suma y resta de dos números binarios representados en módulo y signo. La tabla de la figura 7 da el algoritmo de cálculo. En esta tabla X es el número con signo del acumulador. Y el número con signo del registro de memoria. $|X|$ e $|Y|$ son los módulos de estos números, $|Z|$ el módulo del resultado, x_0, y_0, z_0 los signos de los números antes de la operación y z_0 el signo del resultado. C_0 es el posible resto del orden más elevado del módulo.

Operación	Signos de X e Y x_0 e y_0	Módulo del resultado $\|Z\|$	Signo del resultado z_0 y expresión de $\|Z\|$
Suma	idénticos	$\|Z\| = \|X\| + \|Y\|$	$z_0 = x_0$
	diferentes	$\|Z\| = \|X\| - \|Y\|$	$z_0 = x_0$ si $C_0 = 0$ $z_0 = \bar{x}_0$ si $C_0 = 1$ $\|Z\|$ en complem. a 2
Resta	idénticos	$\|Z\| = \|X\| - \|Y\|$	$z_0 = x_0$ si $C_0 = 0$ $z_0 = \bar{x}_0$ si $C_0 = 1$ $\|Z\|$ en complem. a 2
	diferentes	$\|Z\| = \|X\| + \|Y\|$	$z_0 = x_0$

Fig. 7. *Suma y resta de dos números representados en signo y valor absoluto.*

7.2.2 *Segunda versión: la máquina sólo efectúa adiciones*

En el ejemplo anterior disponíamos de un acumulador que podía efectuar a petición del ordenador una suma o una resta. En muchas máquinas, el acumulador y sus circuitos asociados, sólo pueden efectuar sumas. Los circuitos del ordenador se simplifican. En este caso la máquina procederá de la siguiente forma: el número que se va a sumar o restar X se introduce en el registro de memoria. La máquina comprobará inicialmente si los dos signos de los números son idénticos u opuestos.

Si son idénticos, el ordenador efectuará la suma de los módulos y el signo de los módulos no cambiará. *Si los signos de los dos números son opuestos, la máquina obtiene el complemento a* 1 *del módulo del acumulador cualquiera que sea su contenido, incluso si es positivo. A continuación se suman los módulos.* Si hay un resto C_0 en el orden más elevado del módulo del resultado se suma al del orden más pequeño y se cambia el signo del acumulador cualquiera que fuese su signo anterior. El módulo permanece sin cambiar. Si no existe resto C_0 el signo del acumulador no cambia, pero su módulo debe complementarse a 1 para obtener su

valor real. Los ejemplos que van a continuación demuestran el principio de su funcionamiento.

Ejemplo 1. *Caso de dos números positivos,* $X > 0$, $Y > 0$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	0	0	1	0	0	(+ 4)
Signos idénticos. Adición de módulos		1	0	1	0	
	+	0	1	0	0	
Sin resto más allá del orden más elevado del módulo del resultado. Se conserva el signo del acumulador		1	1	1	0	
Sin modificación del módulo del resultado						
Resultado en el acumulador	0	1	1	1	0	(+ 14)

Ejemplo 2. *Caso de dos números negativos,* $X < 0$, $Y < 0$.

Acumulador (X)	1	1	0	1	0	(− 10)
Registro de memoria (Y)	1	0	1	0	0	(− 4)
Signos idénticos. Suma de módulos		1	0	1	0	
	+	0	1	0	0	
Sin resto superior al orden más elevado del módulo del resultado. Se conserva el signo del acumulador		1	1	1	0	
Sin modificación del módulo del resultado						
Resultado en el acumulador	1	1	1	1	0	(− 14)

Ejemplo 3. *Caso de dos números de signos contrarios,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	1	0	1	0	0	(− 4)
Signos distintos. Complemento a 1 del módulo del acumulador y suma		0	1	0	1	
	+	0	1	0	0	
Sin resto superior al orden más elevado del módulo del resultado. Se conserva el signo del acumulador y se complementa a 1 el módulo		1	0	0	1	
Resultado en el acumulador	0	1	0	0	1	
Resultado en el acumulador después de complementar a 1 el módulo	0	1	1	1	0	(+ 6)

Ejemplo 4. *Caso de dos números de signos opuestos,* $X < 0$, $Y > 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	0	0	1	0	0	(+ 4)
Registro de memoria (Y)	1	1	0	1	0	(− 10)
Signos diferentes. Complemento a 1 del módulo del acumulador y suma		1	0	1	1	
	+	1	0	1	0	
Sin resto más allá del orden más elevado del módulo.		0	1	0	1	
Inversión del signo del acumulador. Sin modificación del					1	
módulo del resultado		0	1	1	0	
Resultado en el acumulador	1	0	1	1	0	

Ejemplo 5. *Caso de dos números de signos opuestos,* $X > 0$, $Y < 0$, *avec* $|X| > |Y|$.

	Sgn					
Acumulador (X)	1	|	1 0 1 0	(– 10)		
Registro de memoria (Y)	0	|	0 1 0 0	(+ 4)		
Signos diferentes: Complemento a 1 del módulo del acumulador y suma			0 1 0 1			
		+	0 1 0 0			
			1 0 0 1			
Sin resto más allá del orden más elevado del módulo Conservación del signo Complemento a 1 del módulo del resultado						
Resultado en el acumulador	1	|	1 0 0 1			
Resultado en el acumulador después de complementar a 1 el módulo	1	|	0 1 1 0	(– 6)		

Ejemplo 6. *Caso de dos números de signos opuestos,* $X < 0$, $Y > 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	1	|	0 1 0 0	(– 4)		
Registro de memoria (Y)	0	|	1 0 1 0	(+ 10)		
Signos opuestos: Complemento a 1 del módulo del acumulador y suma			1 0 1 1			
			1 0 1 0			
			0 1 0 1			
Resto superior al orden más elevado del módulo Inversión del signo del acumulador		+	1			
Sin modificación del módulo del resultado			0 1 1 0			
Resultado en el acumulador	0	|	0 1 1 0	(+ 6)		

Hemos examinado así todos los casos que se producen cuando el ordenador sólo efectúa sumas. La tabla de la figura 8 da el algoritmo en el caso de números binarios expresados en signo y valor absoluto.

Operaciones	x_0, y_0	Módulo del resultado $\|Z\|$	Signo del resultado z_0 y expresiones de $\|Z\|$
La resta se transforma en suma de un número negativo	idénticos	$\|Z\| = \|X\| + \|Y\|$	$z_0 = x_0$ Z en valor real
	diferentes	$\|X\| = \|\overline{X}\|$ $\|Z\| = \|X\| + \|Y\|$	$z_0 = x_0$ si $C_0 = 0$ $\|Z\|$ en complemento a 1 $z_0 = \overline{x_0}$ si $C_0 = 1$ $\|Z\|$ en valor real

Fig. 8. *Algoritmo para el cálculo de la suma y diferencia de dos números binarios expresados en signo y en valor absoluto en el caso en que el ordenador sólo efectúe sumas.*

8. Suma y resta en el caso en que se representan los números negativos por su complemento a 1

8.1 Caso de la suma

Los números negativos se representan por su complemento a 1. Recordemos que el complemento a 1 de un número con signo se obtiene permutando los ceros y los 1. La máquina no se preocupa del signo de los números y efectúa la suma de estos números, el bit de signo se considera como un bit ordinario del módulo. Si existe un resto más allá del signo se suma al orden más bajo del módulo. Si el resultado es positivo (0 en el bit del signo) aparece directamente. Si el resultado es negativo (1 en la posición del signo) aparece bajo la forma de su complemento a 1. Suponemos que el resultado no sobrepasa la capacidad del acumulador.

Ejemplo 1. *Caso de dos números positivos,* $X > 0$, $Y > 0$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	0	0	1	0	0	(+ 4)
Suma de los dos números	0	1	0	1	0	
+ Sin resto más allá del signo	0	0	1	0	0	
El resultado es positivo. Aparece directamente	0	1	1	1	0	
Resultado en el acumulador	0	1	1	1	0	(+ 14)

Ejemplo 2. *Caso de dos números negativos,* $X < 0$, $Y < 0$.

	Sgn					
Acumulador (X)	1	0	1	0	1	(− 10)
Registro de memoria (Y)	1	1	0	1	1	(− 4)
Suma de dos números	1	0	1	0	1	
+	1	1	0	1	1	
Resto más allá del signo	1	0	0	0	0	
+ Suma al orden más pequeño					1	
Resultado negativo	1	0	0	0	1	
Módulo complemento a 1						
Resultado en el acumulador	1	0	0	0	1	
Resultado después del complemento a 1	1	1	1	1	0	(− 14)

Ejemplo 3. *Caso de dos números de signos contrarios,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	1	1	0	1	1	(− 4)
Suma de dos números	0	1	0	1	0	
+	1	1	0	1	1	
Resto más allá del signo	0	0	1	0	1	
+ Suma al orden más pequeño					1	
Resultado positivo. Aparece directamente	0	0	1	1	0	
Resultado en el acumulador.	0	0	1	1	0	(+ 6)

Ejemplo 4. *Caso de dos números de signos contrarios,* $X > 0$, $Y < 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	0	0	1	0	0	(+ 4)
Registro de memoria (Y)	1	0	1	0	1	(– 10)
Suma de dos números	0	0	1	0	0	
Sin resto más allá del signo +	1	0	1	0	1	
Resultado negativo. Módulo complemento a 1	1	1	0	0	1	
Resultado en el acumulador	1	1	0	0	1	
Resultado en el acumulador después del complemento a 1 del módulo	1	0	1	1	0	(– 6)

Ejemplo 5. *Caso de dos números de signos contrarios,* $X < 0$, $Y > 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	1	0	1	0	1	(– 10)
Registro de memoria (Y)	0	0	1	0	0	(+ 4)
Suma de dos números	1	0	1	0	1	
Sin resto más allá del signo +	0	0	1	0	0	
Resultado negativo. Módulo complemento a 1	1	1	0	0	1	
Resultado en el acumulador	1	1	0	0	1	
Resultado en el acumulador después de complementar a 1	1	0	1	1	0	(– 6)

Ejemplo 6. *Caso de dos números de distinto signo,* $X < 0$, $Y > 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	1	1	0	1	1	(– 4)
Registro de memoria (Y)	0	1	0	1	0	(+ 10)
Suma de dos números	1	1	0	1	1	
+	0	1	0	1	0	
Resto más allá del signo	0	0	1	0	1	
Suma al peso más débil +					1	
Resultado positivo. Aparece directamente	0	0	1	1	0	
Resultado en el acumulador	0	0	1	1	0	(+ 6)

Así se han examinado todos los casos de la suma.

En general, la última operación que consiste en convertir un resultado negativo en valor absoluto y en signo no la efectúa la máquina. Esta aparece bajo la forma de su complemento a 1.

8.2 Caso de la substracción

En todos los casos se efectúa la substracción de dos números considerando el bit del signo como un bit ordinario. Si existe un residuo más allá del signo se le resta del orden más elevado del resultado. Si es positivo aparece directamente como complemento a 1. Para obtener el valor real del resultado hay que tomar el complemento a 1 del módulo. Esta última operación generalmente no la efectúa la máquina.

Ejemplo 1. *Caso de dos números positivos*, $X > 0$, $Y > 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	0	0	1	0	0	(+ 4)
Substracción de dos números	0	1	0	1	0	
− Sin residuo más allá del signo	0	0	1	0	0	
Resultado positivo, aparece directamente	0	0	1	1	0	
Resultado en el acumulador	0	0	1	1	0	(+ 6)

Ejemplo 2. *Caso de dos números positivos*, $X > 0$, $Y > 0$, *con* $|X| < |Y|$.

Acumulador (X)	0	0	1	0	0	(+ 4)
Registro de memoria (Y)	0	1	0	1	0	(+ 10)
Substracción de dos números	0	0	1	0	0	
−	0	1	0	1	0	
Residuo más allá del signo	1	1	0	1	0	
− Substracción del peso más débil					1	
Resultado negativo. Módulo complemento a 1	1	1	0	0	1	
Resultado en el acumulador	1	1	0	0	1	
Resultado en el acumulador después de complementar a 1 el módulo	1	0	1	1	0	(− 6)

Ejemplo 3. *Caso de dos números* $X > 0$, $Y > 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	1	0	1	0	1	(− 10)
Registro memoria (Y)	1	1	0	1	1	(− 4)
Substracción de dos números	1	0	1	0	1	
−	1	1	0	1	1	
Residuo más allá del signo	1	1	0	1	0	
− Substracción del peso más débil					1	
Resultado negativo. Módulo complemento a 1	1	1	0	0	1	
Resultado en el acumulador	1	1	0	0	1	
Resultado en el acumulador después de complementar a 1 el módulo	1	0	1	1	0	(− 6)

Ejemplo 4. *Caso de dos números negativos*, $X < 0$, $Y < 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	1	1	0	1	1	(− 4)
Registro de memoria (Y)	1	0	1	0	1	(− 10)
Substracción de dos números	1	1	0	1	1	
− Sin residuo más allá del signo	1	0	1	0	1	
Resultado positivo. Aparece directamente	0	0	1	1	0	
Resultado en el acumulador	0	0	1	1	0	(+ 6)

Ejemplo 5. *Caso de dos números de signos opuestos,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	1	1	0	1	1	(– 4)
Substracción de dos números	0	1	0	1	0	
–	1	1	0	1	1	
Residuo más allá del signo	0	1	1	1	1	
Substracción del peso más débil					1	
Resultado positivo, aparece directamente	0	1	1	1	0	
Resultado en el acumulador	0	1	1	1	0	(+ 14)

Ejemplo 6. *Caso de dos números de signos contrarios,* $X > 0$, $Y < 0$, *con* $|X| < |Y|$.

	Sgn					
Acumulador (X)	0	0	1	0	0	(+ 4)
Registro de memoria (Y)	1	0	1	0	1	(– 10)
Substracción de dos números	0	0	1	0	0	
–	1	0	1	0	1	
Residuo más allá del signo	0	1	1	1	1	
Substracción del peso más débil					1	
Resultado positivo, aparece directamente	0	1	1	1	0	
Resultado en el acumulador	0	1	1	1	0	(+ 14)

Ejemplo 7. *Caso de dos números de signos opuestos,* $X < 0$, $Y > 0$, *con* $|X| > |Y|$.

	Sgn					
Acumulador (X)	1	0	1	0	1	(– 10)
Registro de memoria (Y)	0	0	1	0	0	(+ 4)
Substracción de dos números	1	0	1	0	1	
Sin residuo más allá del signo –	0	0	1	0	0	
Resultado negativo, módulo en complemento a 1	1	0	0	0	1	
Resultado en el acumulador	1	0	0	0	1	
Resultado en el acumulador después de complementar a 1 el módulo	1	1	1	1	0	(– 14)

Ejemplo 8. *Caso de dos números de signos opuestos,* $X < 0$, $Y > 0$, *con* $|X| < |Y|$.

Acumulador (X)	1	1	0	1	1	(– 4)
Registro de memoria (Y)	0	1	0	1	0	(+ 10)
Substracción de los números	1	1	0	1	1	
–						
Sin residuo más allá del signo	0	1	0	1	0	
Resultado negativo. El módulo está en complemento a 1	1	0	0	0	1	
Resultado en el acumulador	1	0	0	0	1	
Resultado después de complementar a 1 el módulo	1	1	1	1	0	(– 14)

Ahora estamos en condiciones de expresar el algoritmo de la suma y la resta de dos números utilizando el complemento a 1 para representar un número negativo (Fig. 9).

Operaciones	Resultado	Observaciones
Adición	$Z = X + Y$	El bit del signo se considera como un bit ordinario. Si existe un residuo más allá del signo se suma en el caso de adición y se resta en el caso de substracción del orden más pequeño del resultado. Si $z_0 = 0$, Z aparece directamente Si $z_0 = 1$, Z está como complemento a 1
Substracción	$Z = X - Y$	

Fig. 9. *Algoritmo de la suma y la resta de dos números utilizando el complemento a 1 para representar un número negativo.*

8.2.1 *Segunda versión: la máquina sólo puede efectuar sumas*

En este caso la substracción de un número se sustituirá por la suma de su complemento a 1. El ordenador sólo efectuará sumas.

Así + 10 − 4 se sustituirá por (+10) + (− 4) y (− 4) se representará por su complemento a 1. Igualmente (− 10) − 4 se representará por (− 10) + (− 4) y los dos números (− 10) y (− 4) se representarán por sus complementos a 1.

Ejemplo 1. *Caso de dos números de signos opuestos,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$ $(10 - 4 = 6)$.

Acumulador (X)		0	1	0	1	0	(+ 10)
Registro de memoria (Y)		1	1	0	1	1	(− 4)
Suma		0	1	0	1	0	
	+	1	1	0	1	1	
Residuo más allá del signo		0	0	1	0	1	
Adición al orden más pequeño	+					1	
Resultado positivo. Aparece directamente		0	0	1	1	0	(+ 6)

Ejemplo 2. *Se desea realizar la siguiente operación:* − 10 − 4 = 14.

Tenemos (− 10) + (− 4) = − 14.

Acumulador (X)		1	0	1	0	1	
Registro de memoria (Y)		1	1	0	1	1	
Suma		1	0	1	0	1	
	+	1	1	0	1	1	
Resto más allá del signo		1	0	0	0	0	
Suma al orden más pequeño	+					1	
		1	0	0	0	1	
Resultado negativo. El módulo es el complemento a 1		1	0	0	0	1	
Resultado en el acumulador después de complementar a 1 el módulo		1	1	1	1	0	(− 14)

Este método tiene la ventaja de prever sólo una suma en programación. La unidad aritmética es mucho más simple. La figura 10 da el algoritmo del cálculo.

Operaciones	x_0, y_0	Resultado	Observaciones
La substracción se transforma en adición de un número complementario a 1.	idénticos	$Z = X + Y$	El bit de signo se considera como un bit ordinario si existe un resto más allá del signo, se suma al orden más inferior del resultado
	diferentes	$Z = X + Y$	Si $z_0 = 0$, Z es el valor real Si $z_0 = 1$, Z está en complemento a 1

Fig. 10. *Algoritmo para el cálculo de una adición y de una substracción de dos números en el caso en que el ordenador sólo realiza adiciones. Los números negativos se expresan mediante sus complementos a 1.*

9. Adición y substracción en el caso en que los números negativos se representan por su complemento a 2

Adición y substracción en el caso en que los números negativos se representan por su complemento a 2.

Este caso es idéntico al del apartado anterior en el que los números negativos se representan por su complemento a 1.

La única diferencia reside en el hecho de que se ignora el residuo más allá del signo. En la figura 11 se da el algoritmo de una adición y una substracción. Se observará que si este resultado es negativo ($z_0 = 1$), el *módulo* del resultado se da en complemento a 2. Los ejemplos que van a continuación dan el método de cálculo. Para una mayor claridad en la exposición los resultados negativos de las operaciones se reconvierten en valor absoluto y en signo. Esta última operación no siempre la efectúa la máquina. El resultado no debe sobrepasar la capacidad del acumulador.

Operaciones	Resultado	Observaciones
Adición	$Z = X + Y$	El bit de signo se considera como un bit ordinario. Se ignora todo resto más allá del signo.
Sustracción	$Z = X - Y$	Si $z_0 = 0$, Z es el valor real Si $z_0 = 1$, Z está en complemento a 2

Fig. 11. *Algoritmo de una adición y una substracción en el caso en que los números negativos se representen mediante su complemento a* 2.

Ejemplo 1. *Adición de dos números negativos,* $X < 0$, $Y < 0$.

		Sgn					
Acumulador (X)		1	0	1	1	0	(− 10)
Registro de memoria (Y)		1	1	1	0	0	(− 4)
Adición de los dos números		1	0	1	1	0	
		1	1	1	0	0	
Se ignora el resto. Resultado negativo	C_0 1	0	0	1	0		
El módulo del resultado está en complemento a 2							
Resultado en el acumulador		1	0	1	1	0	
Resultado en el acumulador después de complementar a 2 el módulo		1	1	1	1	0	(− 14)

Ejemplo 2. *Suma de dos números de signos contrarios,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

		Sgn					
Acumulador (X)		0	1	0	1	0	(+ 10)
Registro de memoria (Y)		1	1	1	0	0	(− 4)
Adición de los dos números		0	1	0	1	0	
Resto ignorado. Resultado positivo		1	1	1	0	0	
Aparece directamente	C_0 0	0	1	1	0		
Resultado en el acumulador		0	0	1	1	0	(+ 6)

Ejemplo 3. *Substracción de dos números negativos,* $X < 0, Y < 0$, *con* $|X| > |Y|$.

		Sgn					
Acumulador (X)		1	0	1	1	0	(− 10)
Registro de memoria (Y)		1	1	1	0	0	(− 4)
Substracción de los dos números		1	0	1	1	0	
		1	1	1	0	0	
Resto ignorado. Resultado negativo. El módulo del resultado está en complemento a 2	C_0 1	1	0	1	0		
Resultado en el acumulador		1	1	0	1	0	
Resultado en el acumulador después de complementar a 2 el módulo		1	0	1	1	0	(− 6)

Ejemplo 4. *Substracción de dos números de signos opuestos,* $X > 0$, $Y < 0$, *con* $|X| > |Y|$.

		Sgn					
Acumulador (X)		0	1	0	1	0	(+ 10)
Registro de memoria (Y)		1	1	1	0	0	(− 4)
Substracción de los dos números		0	1	0	1	0	
		1	1	1	0	0	
Resto ignorado. Resultado positivo. Aparece directamente	C_0 0	1	1	1	0		
Resultado en el acumulador		0	1	1	1	0	(+ 14)

9.1 Caso en que la máquina sólo efectúa adiciones

En este caso se suma el complemento a 2 del número a substraer. Se procede como anteriormente. Se ignorará un resto más allá del signo y el resultado aparecerá directamente si es positivo y en complemento a 2 si es negativo.

Ejemplo 1. *Sea la operación* $10 - 4 = 6$.

	Sgn					
Acumulador (X)	0	1	0	1	0	(+ 10)
Registro de memoria (Y)	1	1	1	0	0	(− 4)
Adición de los dos números	0	1	0	1	0	
	1	1	1	0	0	
Resto ignorado. Resultado positivo. Aparece directamente C_0	0	0	1	1	0	
Resultado en el acumulador	0	0	1	1	0	(+ 6)

La figura 12 da el algoritmo de cálculo en el caso en que la máquina sólo efectúe adiciones.

Operaciones	Resultado	Observaciones
La substracción se transforma en adición complementada a 2 el número que se va a restar	$Z = X + Y$	El bit de signo se considera como un bit ordinario. Se ignora todo resto más allá del signo. Si $z_0 = 0$, Z es el valor real Si $z_0 = 1$, Z está en complemento a 2

Fig. 12. *Algoritmo para la adición y substracción de dos números en el caso en que el ordenador sólo efectúe adiciones. Los números negativos se representan por sus complementos a* 2.

Observación. En el caso de la substracción de dos números idénticos por adición del complemento se obtiene un cero positivo en el caso de complemento a 2 y un cero negativo en el caso de complemento a 1.

10. Tamaño de los números en los registros

La constitución de las memorias y registros en el ordenador nos impone una limitación a los números que puede tratar la máquina. Esta limitación varía según éstos sean mayores o menores que 1 y con el modo de representación de los números negativos.

10.1 Caso de números mayores que 1

Llamemos n el número de dígitos del número incluido el signo. El valor máximo positivo que podemos representar es $(2^{n-1} - 1)$.

Si los números negativos se representan en valor absoluto y en signo, el valor máximo negativo que podemos representar es, naturalmente, $-(2^{n-1} - 1)$. Obtenemos este mismo límite negativo si los números negativos se representan mediante sus complementos a 1.

Tendremos, si es X el número que se va a representar:

$$(2^{n-1} - 1) \geqslant x \geqslant -(2^{n-1} - 1) .$$

Si los números negativos se representan mediante sus complementos a 2 tendremos el mismo límite positivo pero el negativo se aumentará en una unidad, es decir $-(2^{n-1})$. Esto se debe al hecho de no existir el cero negativo en el modo de representación mediante complemento a 2. En efecto, si comparamos los quince primeros números tendremos:

Números	Complementos a 2	
0 | 0 0 0 0	sin cero negativo	1 | 0 0 0 0
0 | 0 0 0 1	1 | 1 1 1 1	
0 | 0 0 1 0	1 | 1 1 1 0	
0 | 0 0 1 1	1 | 1 1 0 1	
.	.	
.	.	
.	.	
0 | 1 0 1 0	1 | 0 1 1 0	
0 | 1 0 1 1	1 | 0 1 0 1	
0 | 1 1 0 0	1 | 0 1 0 0	
0 | 1 1 0 1	1 | 0 0 1 1	
0 | 1 1 1 0	1 | 0 0 1 0	
0 | 1 1 1 1	1 | 0 0 0 1	

Se observará que el valor correspondiente al cero negativo puede ser ocupado por 1 \| 0000 que sucede al último complemento a 2 (se observará el orden decreciente de los módulos). Atribuimos a 1 \| 0000 el valor decimal -16. En estas condiciones los límites de un número x cuyos valores negativos se representan mediante complemento a 2 son:

$$2^{n-1} - 1 \geqslant x \geqslant -2^{n-1}.$$

Ejemplo. *Sumar* -16 a 4.

Obtenemos

$$\begin{array}{l} 1 \mid 0\,0\,0\,0 \quad (-16) \\ \underline{0 \mid 0\,1\,0\,0 \quad (+\ 4)} \\ 1 \mid 0\,1\,0\,0 \quad (-12) \end{array}$$

De acuerdo con lo que acabamos de ver, cada número x deberá estar comprendido en los límites anteriores, *así como el resultado de la operación.* Para poder tratar números importantes se hará n muy grande (20 a 30 por ejemplo), y se introducirá en la máquina números notablemente más pequeños de forma que se tenga a la izquierda de cada número un número suficiente de posiciones disponibles en el registro. No obstante puede ocurrir que el resultado de una operación sobrepase la capacidad máxima de los registros.

Se dice que hay *desbordamiento.* El resto del nivel superior del resultado se incluye en el signo, lo que produce un resultado erróneo.

Ejemplo. *Sea sumar* (+ 10) *a* (+ 7) *en un registro con cinco posiciones comprendido el signo.* Cada uno de estos números no sobrepasa la capacidad máxima de los registros + 15, pero la suma no podrá tener lugar:

$$\begin{array}{l} 0 \mid 1\ 0\ 1\ 0 \quad (+\ 10) \\ \underline{0 \mid 0\ 1\ 1\ 1 \quad (+\ 7)} \\ 1 \mid 0\ 0\ 0\ 1 \end{array}$$

Resultado evidentemente falso. *Se necesita poder detectar este tipo de error para corregir los cálculos.*

Esto se obtiene verificando la existencia de un residuo en el signo y más allá del signo en el caso en que los números negativos se representen por sus complementos a uno o a dos. La tabla de la figura 13 resume todos los casos posibles.

Residuo del módulo en el signo	Residuo más allá del signo	Resultado
sí	sí	exacto
sí	no	fallo
no	sí	fallo
no	no	exacto

Fig. 13. *Test de validez de una suma y una resta (números negativos en complemento a* 1 *o a* 2)

En el caso en que los números se representan en valor absoluto y en signo, se producirá error cuando estemos en presencia de un resto más allá del módulo en el caso de la suma si los signos son idénticos y en el caso de la resta si los signos son diferentes.

10.2 Caso de números más pequeños que 1

En numerosas máquinas los datos son números inferiores a 1. Esta forma de representación ofrece la ventaja de tener la coma situada inmediatamente a la derecha del signo. Esta ventaja es particularmente interesante en el caso de la multiplicación. En efecto, los pesos del resultado que aparecen en el registro Q son los más débiles, los superiores están contenidos en el acumulador. Si no se tiene en cuenta el contenido del registro Q el resultado será aproximado pero suficientemente exacto. Se recordará que en el caso de números superiores a 1, la supresión del contenido del registro conduce a considerables errores. En nuestro caso los

nuevos límites de los números impuestos por las dimensiones de los registros son los siguientes:

$$(1 - 2^{1-n}) \geqslant \times \geqslant -(1 - 2^{1-n})$$

en el caso de una representación en signo y en valor absoluto o en complemento a 1, donde

$$(1 - 2^{1-n}) \geqslant \times \geqslant -1$$

en el caso de una representación de números negativos en complemento a 2. Naturalmente todo lo que se dice anteriormente para los números mayores que 1, en particular lo que concierne a los tests de validez del resultado permanece válido en el caso de números inferiores a 1.

11. Desplazamiento de los números en los registros

Un número, una vez introducido en un registro, puede experimentar ciertas operaciones de traslado que se llaman desplazamiento a la derecha y a la izquierda. Un número puede desplazarse un nivel o varios niveles a la derecha o a la izquierda. Si el número es positivo un desplazamiento a la izquierda de 1, 2,..., n niveles, tiene como efecto el *multiplicar* este número por $2^1, 2^2, \ldots, 2^n$.

Igualmente un desplazamiento a la derecha del número binario de 1, 2, ..., n niveles tiene como efecto el dividir el valor decimal de este número por $2^1, 2^2, \ldots, 2^n$. En los dos casos se añadirán a la izquierda del número tantos ceros como desplazamientos haya habido.

Ejemplo. *Desplazar el equivalente binario de* 10 *de* 1 *y* 2 *niveles.*

2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	
	1	0	1	0					(+ 10)

Desplazamiento de un nivel a la izquierda

1	0	1	0	0	0				(+ 20)

Se añade un cero a la derecha del número

Desplazamiento de un nivel a la derecha.

0	0	1	0	1	0				(+ 5)

Se añade un cero a la izquierda del número.

En el caso de números negativos el desplazamiento se hace más complicado. En todos los casos deberá conservarse el signo y el desplazamiento se efectuará únicamente sobre el módulo del número. En lo que va a continuación supondremos que el número dispone de una o varias posiciones a su derecha y a su izquierda. Examinaremos primeramente el caso de un número negativo (– 10) expresado en valor absoluto y en signo.

11.1 Desplazamiento de un número negativo expresado en valor absoluto y en signo

Desplazamiento a izquierda		Desplazamiento a derecha	
(– 10) 1 | | 1 0 1 0		1 | 1 0 1 0 |	(– 10)
(– 20) 1 | 1 | 0 1 0 0		1 | 0 1 0 1 | 0	(– 5)
(– 40) 1 | 1 0 | 1 0 0 0		1 | 0 0 1 0 | 1 0	(– 2.5)

Se observará que en los dos casos se añade a la derecha o a la izquierda del módulo un número de ceros igual al de desplazamientos. Si el registro contiene inicialmente cuatro niveles más el signo, los dígitos a la izquierda y a la derecha de la línea punteada se perderán. Examinemos ahora el caso de un número negativo expresado mediante su complemento a 2.

11.2 Desplazamiento de un número negativo expresado mediante su complemento a 2

Desplazamiento a izquierda		Desplazamiento a derecha	
(– 10) 1 | | 0 1 1 0		1 | 0 1 1 0 |	(– 10)
(– 20) 1 | 0 | 1 1 0 0		1 | 1 0 1 1 | 0	(– 5)
(– 40) 1 | 0 1 | 1 0 0 0		1 | 1 1 0 1 | 1 0	(– 2.5)

Se observa que en caso de un desplazamiento a la izquierda se añaden los ceros a la derecha o a la izquierda del módulo. Si el registro se limita a cuatro posiciones para el módulo, se pierden los dígitos a la izquierda y a la derecha de la línea punteada. Veamos ahora el último caso, el de un número negativo que se expresa por su complemento a 1.

11.3 Desplazamiento de un número negativo expresado por su complemento a 1

Volvamos al ejemplo anterior y desplacemos el número (– 10) dos niveles.

Desplazamiento a la izquierda		Desplazamiento a la derecha	
(– 10) 1 | | 0 1 0 1		1 | 0 1 0 1 |	(– 10)
(– 20) 1 | 0 | 1 0 1 1		1 | 1 0 1 0 | 1	(– 5)
(– 40) 1 | 0 1 | 0 1 1 1		1 | 1 1 0 1 | 0 1	(– 2.5)

Observaremos esta vez que el número a desplazar se representa por su complemento a 1, se añaden a la derecha y a la izquierda tantos 1 como desplazamientos. La tabla de la figura 14 resume lo que acabamos de ver.

Naturaleza del número	Signo	Observaciones
Número positivo	sin cambiar	Añadir ceros a la derecha o a la izquierda del número
Número negativo representado en valor absoluto y en signo	sin cambiar	Añadir ceros a la derecha o a la izquierda del módulo del número
Número negativo en complemento a 2	sin cambiar	Desplazamiento a la izquierda: añadir ceros Desplazamiento a la derecha: añadir 1
Número negativo en complemento a 1	sin cambiar	Añadir 1 a derecha o a izquierda del módulo de número

Fig. 14. *Algoritmo de desplazamiento de un número binario.*

12. La multiplicación en las calculadoras. Generalidades

La regla de la multiplicación binaria es la misma que la utilizada en la multiplicación manual. Se multiplica cada dígito del multiplicador por el multiplicando. Cada producto parcial obtenido de esta manera será o 0 o el propio multiplicando. Si el multiplicando tiene m dígitos y el multiplicador n dígitos el resultado tendrá como máximo $m + n$ dígitos.

Habrá n productos parciales que se sumarán dos a dos después de efectuar los correspondientes desplazamientos. La suma obtenida de esta manera representa la sucesión de las sumas parciales.

12.1 Ejemplo de una multiplicación manual. Desplazamiento a la izquierda

Examinemos primeramente el caso de una multiplicación binaria realizada manualmente, sea efectuar el producto $3_{10} \times 6_{10} = 18_{10}$. Tenemos:

```
  0 1 1     X Multiplicando (3)
× 1 1 0     Y Multiplicador (6)
  0 0 0
0 1 1
0 1 1
1 0 0 1 0       Resultado de (18)
```

Se multiplica primeramente el dígito de nivel inferior del multiplicador por el multiplicando. En nuestro caso el multiplicando es 011 y el multiplicador es 110. El dígito de nivel inferior del multiplicador es 0 que multiplicado por el multiplicando da el primer producto parcial 0000. El segundo dígito del multiplicador es 1 que multiplicado por el multiplicando da el segundo producto parcial 110. Se observa que este producto parcial es igual al multiplicando. Se desplaza a la izquierda una posición todo el producto parcial. A continuación se realiza el tercer producto parcial que se desplaza nuevamente una posición hacia la izquierda. La suma de todos estos productos parciales da el resultado de la multiplicación $18_{10} = 10010$.

12.2 Multiplicación automática

En el caso de la multiplicación automática no se espera a tener el resultado de todos los productos parciales para efectuar la suma total. Esta se obtiene realizando la suma parcial del producto parcial precedente y del producto parcial siguiente desplazado un nivel a la izquierda. Así, siguiendo este método, obtenemos para el ejemplo anterior:

$$
\begin{array}{rl}
011 & \\
110 & \\
\hline
000 & \\
011 & \\
\hline
0110 & \text{primera suma parcial} \\
011 & \\
\hline
10010 & \text{segunda suma parcial}
\end{array}
$$

De una forma más general, llamando X al multiplicando, Y al multiplicador, y_n, y_{n-1}, ..., y_0 los dígitos de las diferentes órdenes del multiplicador, $P_0, P_1, ..., P_n$ los diferentes productos parciales de cada dígito del multiplicador por el multiplicando, 2^0, 2^1, ..., 2^n un factor que indica que se efectúa un desplazamiento hacia la izquierda 0, 1, ..., n niveles de las cantidades a las que se le asocia. Tomando el ejemplo anterior tenemos la siguiente tabla (Fig. 15) que da todas las etapas de la multiplicación. Lo que puede representarse por la relación más general.

$$XY = Xy_0 + Xy_1\, 2^1 + \cdots + Xy_n\, 2^n$$

$$XY = P_0 + P_1 + \cdots + P_n\,.$$

Multiplicando X Multiplicador Y		0 1 1 1 1 0 $y_2\ y_1\ y_0$	Observaciones
$P_0 = Xy_0\ 2^0$		0 0 0	Primer producto parcial
$P_1 = Xy_1\ 2^1$	0	1 1	Segundo producto parcial desplazado un nivel a la izquierda
$S_1 = P_0 + P_1$	0	1 1 0	Primera suma parcial
$P_2 = Xy_2\ 2^2$	0 1	1	Tercer producto parcial desplazado 2 niveles a la izquierda
$S_2 = S_1 + P_2 = P_0 + P_1 + P_2$	1 0	0 1 0	Segunda suma parcial

Fig. 15. *Diferentes fases de la multiplicación* 011 x 110. *Desplazamiento a la izquierda.*

Se observará que cada producto parcial se desplaza antes de sumarse, es decir que entra en el registro de suma en diferentes posiciones según su nivel en la operación a realizar. Esto es un inconveniente pues la adición de sumas parciales se efectúa sobre ocho posiciones en dos registros puestos en serie. Se remedia este inconveniente utilizando el desplazamiento a la derecha para realizar la multiplicación. En este método las sumas parciales se desplazan un nivel a la derecha. Los productos parciales no experimentan desplazamiento.

Multiplicando X Multiplicador Y	1 1 1 1 0 1 $y_2\ y_1\ y_0$		Observaciones
Suma anterior	0 0 0		Suma anterior
$P_0 = Xy_0$	1 1 1		Primer producto parcial
S_0	1 1 1		Primera suma parcial
$S_0\ 2^{-1}$	0 1 1	1	Primera suma parcial desplazada
$P_1 = Xy_1$	0 0 0		Segundo producto parcial
$S_1 = S_0\ 2^{-1} + Xy_1$	0 1 1	1	Segunda suma parcial
$S_1\ 2^{-1}$	0 0 1	1 1	Segunda suma parcial desplazada
$P_2 = Xy_2$	1 1 1		Tercer producto parcial
$S_2 = S_1\ 2^{-1} + Xy_2$	1 0 0 0	1 1	Tercera suma parcial
$S_2\ 2^{-1}$	0 1 0 0	0 1 1	Tercera suma parcial desplazada (Resultado)

Fig. 16. *Diferentes etapas de la multiplicación de* 111 *por* 101. *Desplazamiento a la derecha.*

La figura 16 da las diferentes etapas de la multiplicación de 111 por 101. El operador 2^{-1} indica un desplazamiento de un nivel a la derecha. Se observará que la suma se realiza sobre tres niveles. La segunda columna resulta únicamente del desplazamiento de las sumas parciales. El desplazamiento de la derecha da el mismo resultado que un desplazamiento a la izquierda. Así utilizando el desplazamiento a la izquierda en el ejemplo anterior

```
    1 1 1      X (7)
    1 0 1      Y (5)
    -----
    1 1 1
  0 0 0
1 1 1
---------
1 0 0 0 1 1    Resultado (35)
```

obtenemos el mismo resultado. No obstante, un desplazamiento a la derecha es más fácil de realizar por las razones enumeradas anteriormente. Un multiplicador se realiza mediante tres registros y un adicionador. Si los números a multiplicar tienen posiciones el acumulador tendrá $n + 1$ posiciones para tener en cuenta a un posible resto. Contendrá una vez realizada la operación los niveles más altos del resultado. El problema de desbordamiento de capacidad no existe en la multiplicación.

13. El problema de los signos

En la mayor parte de los multiplicadores los números negativos se introducen mediante un módulo igual al valor absoluto del número y un 1 en la posición del signo. Así – 5 se representará mediante 1.101 y – 10 mediante – 1.1010. Los signos del multiplicando y multiplicador no participan en la elaboración de los productos parciales y se tratan separadamente. En particular el signo del multiplicador se registra en un circuito de doble posición. Llamemos signo 1, signo 2 y signo el signo del multiplicador, del multiplicando y del resultado. Obtenemos la siguiente tabla de verdad atribuyendo el valor 0 al signo + y el 1 al signo – (Fig. 17):

Sgn 1	Sgn 2	Sgn
0	0	0
0	1	1
1	0	1
1	1	0

Fig. 17. *Multiplicación de los signos en el multiplicador.*

Obtenemos:

$$\text{Sgn} = \overline{\text{Sgn 1}}\ \text{Sgn 2} + \text{Sgn 1}\ \overline{\text{Sgn 2}}\ ,$$

o

$$\text{Sgn} = \text{Sgn 1} \oplus \text{Sgn 2}\ .$$

El signo del resultado es el "O exclusivo" de los signos del multiplicador y del multiplicando. Este se obtiene, según hemos visto en el capítulo 1, mediante el esquema de la figura 18. Sgn 1 representa el signo del multiplicador, Sgn 2 el signo del multiplicando. Se opera de la siguiente forma: se introduce el multiplicador y el multiplicando en sus respectivos registros. Las básculas del signo fijan Sgn 1 y Sgn 2. Se aplica a continuación impulso Prod Sgn sobre el circuito Y que presenta la salida verdadera de la báscula Sgn 2 a la entrada de la báscula Sgn 1. Esta fija entonces el signo del resultado según la tabla de la figura 18. *Así se efectúa el producto de los signos antes de empezar la multiplicación.*

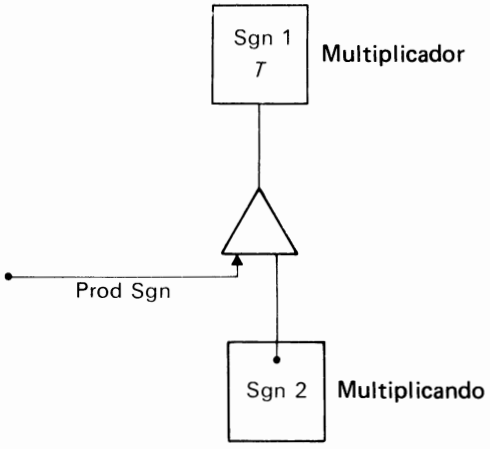


Fig. 18. *Realización del producto de los signos en el multiplicador.*

14. Desarrollo del algoritmo de la multiplicación de dos números expresados en valor absoluto y en signo

Llamemos X al multiplicando, Y al multiplicador de dígitos $y_0, \ldots, y_n$ y Z el producto XY. Suponemos que la representación de los números es inferior a 1. El método es general y es válido para los números superiores a 1.

Los números se representan de la siguiente forma

$$\begin{array}{cc} & 2^0 \mid 2^{-1} \ldots\ldots 2^{-n} \\ X & x_0 \mid x_1 \ldots\ldots x_n \\ Y & y_0 \mid y_1 \ldots\ldots y_n \end{array} \qquad (1)$$

la máquina tratará separadamente los signos x_0 e y_0 de X y de Y. Llamemos

$$| Z | = | X | | Y |$$

y z_0 el signo del resultado Z. Obtenemos:

$$z_0 = x_0 \oplus y_0 \qquad (2)$$

la máquina elabora el producto de los módulos

$$| Z | = | X | | Y | = | X | (y_1 2^{-1} + y_2 2^{-2} \ldots y_n 2^{-n}) . \qquad (3)$$

Llamemos S_n la suma parcial de orden n obtenida sumando la suma parcial anterior desplazada un nivel a la derecha (multiplicado por 2^{-1}) y el producto parcial de nivel n, $| X | y_n$:

$$S_n = S_{n+1} 2^{-1} + | X | y_n . \qquad (4)$$

La suma parcial corresponde al dígito de orden $n - p$ del multiplicador es

$$S_{n-p} = S_{n-p+1} 2^{-1} + | X | y_{n-p} \qquad (5)$$

donde p varía de 0 a $n - 1$. Para $p = 0$ obtenemos la expresión (4). Al comienzo de la operación la suma parcial precedente es nula, de donde

$$S_n = | X | y_n \quad (p = 0) . \qquad (6)$$

Obtenemos a continuación sucesivamente

$$S_{n-1} = S_n 2^{-1} + | X | y_{n-1} = | X | y_n 2^{-1} + | X | y_{n-1} \qquad (p-1)$$

$$S_{n-2} = S_{n-1} 2^{-1} + | X | y_{n-2} = | X | y_n 2^{-2} + | X | y_{n-1} 2^{-1} + | X | y_{n-2} \qquad (p=2)$$

$$\vdots$$

$$S_1 = | X | y_n 2^{-(n-1)} + | X | y_{n-1} 2^{-(n-2)} + \cdots + y_1 | X | \qquad (p=n-1)$$

Multipliquemos miembro a miembro por 2^{-1}:

$$S_1 2^{-1} = | X | (y_1 2^{-1} + y_2 2^{-2} + \cdots + y_n 2^{-n}) = | X | | Y | . \qquad (7)$$

La tabla (Fig. 19) da el algoritmo de la multiplicación de dos números representados en valor absoluto y en signo. Si los números a multiplicar son mayores que 1:

$$\begin{aligned} &2^n \mid 2^{n-1} \ldots\ldots 2^0 \\ &x_n \mid x_{n-1} \ldots\ldots x_0 \\ &y_n \mid y_{n-1} \ldots\ldots y_0 \,. \end{aligned} \tag{8}$$

Observaremos que x_n e y_n son los signos de X e Y:

$$z_n = x_n \oplus y_n \tag{9}$$

donde z_n es el signo del producto Z. La suma parcial de orden k es

$$S_k = S_{k-1}\, 2^{-1} + |X|\, y_k \quad (k = 0, \ldots, n-1)\,. \tag{10}$$

Dando a k todos los valores comprendidos entre 0 y $n-1$ obtenemos $(S_{-1} = 0)$:

$$S_0 = |X|\, y_0 \qquad (k = 0)$$

$$S_1 = S_0\, 2^{-1} + |X|\, y_1 = |X|\, y_0\, 2^{-1} + |X|\, y_1 \qquad (k = 1)$$

$$S_2 = S_1\, 2^{-1} + |X|\, y_2 = |X|\, y_0\, 2^{-2} + |X|\, y_1\, 2^{-1} + |X|\, y_2 \qquad (k = 2)$$

$$\vdots$$

$$\begin{aligned} S_{n-1} = S_{n-2}\, 2^{-1} + xy_{n-1} &= |X|\, y_0\, 2^{-(n-1)} \\ &+ |X|\, Y_1\, 2^{-(n-2)} + \cdots + |X|\, y_{n-1} \end{aligned} \qquad (k = n-1)$$

Multipliquemos miembro a miembro por 2^{n-1}:

$$S_{n-1}\, 2^{n-1} = X(y_{n-1}\, 2^{n-1} + Y_{n-2}\, 2^{n-2} + \cdots + y_0\, 2^0) = |X|\,|Y|\,. \tag{11}$$

El algoritmo es similar al de los números inferiores a 1. Igualmente se da en la figura 19.

Operaciones	Números inferiores a 1 $2^0\ 2^{-1} \ldots 2^{-n}$	Números superiores a 1 $2^n\ 2^{n-1} \ldots 2^0$
Signo del producto	$z_0 = x_0 \oplus y_0$	$z_n = x_n \oplus Y_n$
Suma parcial de orden p, n y orden k	$S_{n-p} = S_{n-p+1}\, 2^{-1} + \lvert X \rvert\, y_{n-p}$ $(p = 0, \ldots, n-1)$	$S_k = S_{k-1}\, 2^{-1} + \lvert X \rvert\, y_n$ $k = 0, \ldots, n-1$
	$S_n = \lvert X \rvert\, y_n \qquad S_{n+1} = 0$	$S_0 = Xy_0 \qquad S_{-1} = 0$
Módulo del producto	$\lvert Z \rvert = \lvert X \rvert\,\lvert Y \rvert = S_1\, 2^{-1}$	$S_{n-1}\, 2^{n-1} = \lvert X \rvert\,\lvert Y \rvert = \lvert Z \rvert$

Fig. 19. *Algoritmo de la multiplicación de dos números expresados en valor absoluto y en signo.*

15. Multiplicación por el método de Booth

En este método los números negativos se representan *mediante su complemento a 2 y la multiplicación se efectúa considerando el bit del signo como un bit ordinario.* Es, en suma, el equivalente del complemento a 2 en la adición. La máquina examina sucesivamente los niveles significativos del multiplicador tomados dos a dos y comenzando por el menor valor. Según las cuatro posibilidades 00, 11, 10 y 01 efectúa las operaciones indicadas en la figura 20.

Los desplazamientos a la derecha del contenido del acumulador y del registro se efectúan conservando el signo del acumulador, salvo en el caso en que el multiplicando es igual al máximo valor negativo 10000 = – 16 que conduce, eventualmente, a un desbordamiento de la capacidad del acumulador. El desplazamiento se efectúa invirtiendo el signo del acumulador.

Bits del multiplicador	Operaciones
00	Desplazar el producto parcial un orden a la derecha
01	Sumar el multiplicando, después desplazar un orden a la derecha el producto parcial
10	Restar el multiplicando, después desplazar un orden a la derecha el producto parcial
11	Desplazar el producto parcial un orden a la derecha

Fig. 20. *Algoritmo de la multiplicación por el método de Booth.*

Antes de comenzar la multiplicación se añade un cero a la derecha del multiplicador. El elemento multiplicador de Booth lleva tres registros + más un esquema de dos posiciones que toma los dígitos sucesivos de orden más pequeño del multiplicador y un dispositivo de comparación de los dos últimos órdenes del multiplicador. Un contador cuenta el número de desplazamientos y detiene la multiplicación una vez que éstos han terminado.

El multiplicando X se introduce en un registro multiplicando, el multiplicador se introduce en el registro Q. Este se dispone en serie con el acumulador. El signo del acumulador y del multiplicador experimentan los desplazamientos pues se considera como un dígito ordinario de los dos números. Si el resultado de la multiplicación es positivo aparece directamente, si es negativo mediante su complemento a 2.

Ejemplo 1. *Sea multiplicar* $X = (+10)$ *por* $(Y = -2)$.

X es el multiplicando, Y el multiplicador.
Y se representa mediante su complemento a 2.

$Q_5\ Q_S$	Microoperaciones	Acumulador	Registro $Q\ Q_5$	Q_S	C
		0 0 0 0 0	1 1 1 1 0	0	5
00	* Desp. dcha.	0 0 0 0 0	0 1 1 1 1	0	4
10	Substracción de X	− 0 1 0 1 0			
		1 0 1 1 0	0 1 1 1 1	0	
	Desp. dcha.	1 1 0 1 1	0 0 1 1 1	1	3
11	Desp. dcha.	1 1 1 0 1	1 0 0 1 1	1	2
11	Desp. dcha.	1 1 1 1 0	1 1 0 0 1	1	1
11	Desp. dcha (Resultado)	1 1 1 1 1	0 1 1 0 0	1	0

* Desp. dcha.: desplazamiento a la derecha.

Ejemplo 2. *Sea multiplicar* (-11) *por* (-55).

$X = (-11)$ es el multiplicando.
$Y = (-5)$ el multiplicador. El resultado es $+55$. Los dos operandos son negativos y se representan mediante sus complementos a 2. Tenemos entonces:

$Q_5\ Q_S$	Microoperaciones	Acumulador	Registro $Q\ Q_5$	Q_S	C
		0 0 0 0 0	1 1 0 1 1	0	5
10	Substracción de X	− 1 0 1 0 1			
		0 1 0 1 1	1 1 0 1 1	0	
	Desp. dcha.	0 0 1 0 1	1 1 1 0 1	1	4
11	Desp. Dcha.	0 0 0 1 0	1 1 1 1 0	1	3
01	Suma de X	+ 1 0 1 0 1			
		1 0 1 1 1	1 1 1 1 0	1	
	Desp. dcha.	1 1 0 1 1	1 1 1 1 1	0	2
10	Substracción de X	− 1 0 1 0 1			
		0 0 1 1 0	1 1 1 1 1	0	
	Desp. dcha.	0 0 0 1 1	0 1 1 1 1	1	1
11	Desp. dcha. (Resultado)	0 0 0 0 1	1 0 1 1 1	1	0

Ejemplo 3. Caso en que el multiplicando tiene el valor negativo máximo.

$$X = 10000\ (-16)\quad Y = 11011\ (-5).$$

Desbordamiento de la capacidad del acumulador.

$Q_5\ Q_S$	Microoperaciones	Acumulador	Registro $Q\ Q_5$	Q_S	C
		0 0 0 0 0	1 1 0 1 1	0	5
	Substracción de X	− 1 0 0 0 0			
10	Desbordamiento de capacidad	1 0 0 0 0	1 1 0 1 1	0	
	Desp. dcha.	[0]1 0 0 0	[0]1 1 0 1	1	4
11	Desp. dcha.	0 0 1 0 0	[0 0]1 1 0	1	3
	Suma de X	+ 1 0 0 0 0			
01		1 0 1 0 0	[0 0]1 1 0	1	
	Desp. dcha.	1 1 0 1 0	[0 0 0]1 1	0	2
	Substracción de X	− 1 0 0 0 0			
10		0 1 0 1 0	0 0 0 1 1	0	
	Desp. dcha.	0 0 1 0 1	[0 0 0 0]1	1	1
11	Desp. dcha. (Resultado)	[0 0 0 1 0]	[1 0 0 0 0]	1	0

Se observará el desbordamiento de capacidad del acumulador desde la primera operación y el desplazamiento con cambio de signo.

15.1 Justificación de la multiplicación mediante el método de Booth

Llamemos X al multiplicando, Y al multiplicador, $y_o, y_1, \ldots, y_n$ cada uno de los dígitos del multiplicador, el dígito y_0 corresponde al signo. Tenemos la siguiente disposición, suponiendo una representación de números inferior a 1.

$$2^0 \quad 2^{-1} \ldots\ldots 2^{-n}$$

$$y_0 \quad y_1 \cdot \ldots\ldots y_n \quad y_{n+1}\ .$$

Según el método descrito anteriormente tenemos, para $n \neq 0$:

$y_n = y_{n+1}$	Desplazamiento a la derecha un orden el contenido del acumulador.
$y_n = 1, y_{n+1} = 0$	Substraer X al contenido del acumulador, después de desplazar el resultado un orden a la derecha.
$y_n = 0, y_{n+1} = 1$	Sumar X al contenido del acumulador, después desplazarlo un orden a la derecha.

Llamemos a_n el estado del acumulador al comienzo de la multiplicación y $a_n - p$ su contenido después de p operaciones elementales, p varía desde 0 a n. Obtenemos:

$$a_{n-p} = \tfrac{1}{2}\, a_{n-p+1} + (y_{n-p+1} - y_{n-p})\, X \qquad (p = 0, \ldots, n)\,.$$

Al comienzo de la multiplicación $p = 0$, de donde

$$a_n = \tfrac{1}{2}\, a_{n+1} + (y_{n+1} - y_n)\, X\,.$$

Pero en este momento el contenido anterior del acumulador es nulo, así como y_{n+1} que está impuesto por el método:

$$a_n = -\, y_n\, X\,, \quad p = 0\,.$$

Obtenemos entonces, multiplicando miembro a miembro la expresión general por 2^p:

$$2^p\, a_{n-p} = 2^{p-1}\, a_{n-p+1} + 2^p\, (y_{n-p+1} - y_{n-p})\, X \quad (p = 0, \ldots, n)$$

para los diferentes valores de p tenemos:

$$\begin{aligned}
&a_n = -\, y_n\, X && p = 0\\
&2\, a_{n-1} = a_n + 2(y_n - y_{n-1})\, X && p = 1\\
&2^2\, a_{n-2} = 2\, a_{n-1} + 2^2(y_{n-1} - y_{n-2})\, X && p = 2\\
&2^3\, a_{n-3} = 2^2\, a_{n-2} + 2^3(y_{n-2} - y_{n-3})\, X && p = 3\\
&\vdots && \vdots\\
&2^{n-1}\, a_1 = 2^{n-2}\, a_2 + 2^{n-1}(y_2 - y_1)\, X && p = n - 1\\
&2^n\, a_0 = 2^{n-1}\, a_1 + 2^n(y_1 - y_0)\, X && p = n\,.
\end{aligned}$$

Sumemos miembro a miembro y obtendremos:

$$2^n\, a_0 = [y_n + 2\, y_{n-1} + 2^2\, y_{n-2} + \cdots + (2^n - 2^{n-1})\, y_1 - 2^n\, y_0]\, X$$

dividiendo por 2^n:

$$a_0 = (-\, y_0 + 2^{-1}\, y_1 + \cdots + 2^{-(n-2)}\, y_{n-2} + 2^{-(n-1)}\, y_{n-1} + 2^{-n}\, y_n)\, X\,;$$

a_0 representa el estado final del conjunto acumulador - registro Q, es decir el producto XY:

$$a_0 = XY = \left(-\, y_0 + \sum_{k=1}^{n} 2^{-k}\, y_k\right) X\,.$$

La cantidad entre paréntesis no es otra cosa que el multiplicador y:

$$y = -\, y_0 + \sum_{k=1}^{n} 2^{-k}\, y_k\,.$$

Finalmente el estado del acumulador y del registro Q después de p operaciones elementales, es decir, después de haber agotado todas las combinaciones del multiplicador, incluido el signo es:

$$a_0 = XY ,$$

que corresponde al resultado de la multiplicación.

Este método expone el resultado con el signo conveniente.

15.2 Variante del método de Booth

En este método, muy parecido al anterior, se examina el orden inferior del registro Q. Según la naturaleza de este orden y de la operación anterior se efectuará una suma o una resta seguida de un desplazamiento o un desplazamiento sin operación. *Al comienzo de la operación, se supone que la operación anterior era una adición.*

Si el orden inferior del registro Q es 0 no se efectúa ninguna operación y se desplaza un orden a la derecha el contenido del acumulador y del registro Q hasta la aparición de un 1 en el orden inferior del registro Q. En este momento se examina el segundo orden inferior del registro Q y se efectúan las operaciones indicadas en la tabla de la fig. 21. Si en el *transcurso de la operación,* el último orden del registro Q es 1 y la operación anterior ha sido una *substracción* se desplaza el contenido del acumulador y del registro Q un orden a la derecha hasta la aparición de un cero en el último orden del registro Q. Se examina entonces el segundo orden de menos valor y se efectúan las operaciones indicadas en la tabla de la figura 21. Se procede así por aproximaciones hasta el momento en que el orden superior del multiplicador aparezca en el lugar del orden inferior del registro Q. El penúltimo orden de este registro contiene entonces una parte del resultado de la multiplicación: no puede tomarse en consideración. Se procede entonces de la siguiente manera: si la operación anterior es una suma y el orden más inferior del registro Q es 1 se efectuará una substracción del multiplicando del contenido del acumulador. Si la operación anterior es una substracción y el orden más inferior del registro Q es 0 y se efectúa una suma del multiplicando al contenido del acumulador.

Esta última operación va seguida siempre de un desplazamiento. En todos los otros casos la operación consistirá en un simple desplazamiento. La *multiplicación* se termina entonces y el resultado aparece en el acumulador y el registro Q. Este método se justifica porque el multiplicador puede obtenerse siempre mediante una sucesión de suma y resta seguida de un desplazamiento. Así, el multiplicador:

$$Y = 1\,0\,1\,1\,1\,0\,1\,0\,0\,0$$

puede reconstituirse de la siguiente forma.

$$
\begin{array}{ll}
Y & 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0 \\
\hline
 & 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0 \\
 & +\qquad\qquad\qquad 1 \\
\hline
 & 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0 \\
 & +\qquad\qquad 1 \\
\hline
 & 1\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0 \\
 & -1 \\
\hline
 & 1\ 0\ 1\ 1\ 1\ 0\ 1\ 0\ 0\ 0
\end{array}
$$

Bits del multiplicador	Operación anterior	Nueva operación
0	Suma	Desplazamiento a la derecha
01		Adición, desplazamiento a la derecha
11		Substracción, desplazamiento a la derecha
00	Substracción	Adición, desplazamiento a la derecha
10		Substracción, desplazamiento a la derecha
1		Desplazamiento a la derecha

Fig. 21. *Variante del método de Booth.*

Veamos algunos ejemplos:

Ejemplo 1. *Sea multiplicar el multiplicando* $X = 1\ 0\ 1\ 0\ 0$ (− 12) *por el multiplicador* $Y = 0\ 1\ 0\ 1\ 0$ (+ 10).

Tenemos las siguientes operaciones. 0 representa el estado del contador de desplazamientos (Fig. 22). El resultado es:

$$XY = 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ (-\ 120)\ .$$

$Q_4\ Q_5$	Operación anterior	Nueva operación	Acumulador	Registro Q	C
			0 | 0 0 0 0	0 | 1 0 1 0	5
0	Suma	Despl. dcha.	0 | 0 0 0 0	0 | 0 1 0 1	4
01	Suma	Suma X	1 | 0 1 0 0		
			1 | 0 1 0 0	0 | 0 1 0 1	4
		Desp. dcha.	1 | 1 0 1 0	0 | 0 0 1 0	3
0	Suma	Desp. dcha.	1 | 1 1 0 1	0 | 0 0 0 1	2
01	Suma	Suma X	1 | 0 1 0 0		
			1 | 0 0 0 1	0 | 0 0 0 1	2
		Desp. dcha.	1 | 1 0 0 0	1 | 0 0 0 0	1
0	Suma	Desp. dcha. (Resul.)	1 | 1 1 0 0	0 | 1 0 0 0	0

Fig. 22. *Detalle de la multiplicación del ejemplo* 1.

El resultado $XY = 1\ 1\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0$ representa $-$ 120 en complemento a 2.

Ejemplo 2. *Sea multiplicar el multiplicando* $X = 0\ 1\ 0\ 1\ 0$ (+ 10) *por el multiplicador* $Y = 1\ 1\ 1\ 1\ 0$ (−2).
La figura 23 da el detalle la multiplicación. El resultado es

$$XY = 1\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 0\ 0 \quad (-20).$$

$Q_4\ Q_5$	Operación anterior	Nueva operación	Acumulador	Registro Q	C
			0 | 0 0 0 0	1 | 1 1 1 0	5
0	Suma	Desp. dcha.	0 | 0 0 0 0	0 | 1 1 1 1	4
11	Suma	Substracción X	0 | 1 0 1 0		
			1 | 0 1 1 0	0 | 1 1 1 1	4
		Desp. dcha.	1 | 1 0 1 1	0 | 0 1 1 1	3
1	Substracción	Desp. dcha.	1 | 1 1 0 1	1 | 0 0 1 1	2
1	Substracción	Desp. dcha.	1 | 1 1 1 0	1 | 1 0 0 1	1
1	Substracción	Desp. dcha. (Resul.)	1 | 1 1 1 1	0 | 1 1 0 0	0

Fig. 23. *Detalle de la multiplicación del ejemplo* 2.

Ejemplo 3. *Multiplicando* $X = 10011$ (-13)
Multiplicador $Y = 00101$ $(+5)$.

$Q_4\ Q_5$	Operación anterior	Nueva operación	Acumulador	Registro Q	C
			00000	00101	5
01	Adición	Adición	+ 10011		
			10011	00101	
		Desp. dcha.	11001	10010	4
0	Adición	Desp. dcha.	11100	11001	3
01	Adición	Adición	+ 10011		
		Desbordamiento de cap.	01111	11001	
		Desp. dcha.	10111	11100	2
0	Adición	Desp. dcha.	11011	11110	1
0	Adición	Desp. dcha. (Resultado)	11101	11111	0

En este ejemplo se observará un desbordamiento de capacidad del acumulador desde la segunda adición. El desplazamiento que se efectúa a continuación se hace con inversión del signo del acumulador. La determinación del signo puede igualmente obtenerse una vez realizado el desplazamiento, lo que necesita, naturalmente, un tiempo elemental suplementario. Considerando los ejemplos anteriores y efectuando otros ejemplos que supongan un desbordamiento de capacidad del acumulador durante la multiplicación, se observará que la posición del signo después de cada desplazamiento se determina en todos los casos por la siguiente tabla (Tabla 1). Esta tabla es válida igualmente para el primer algoritmo de Booth.

Operación	Signo del multiplicando	Signo de A_0
Adición	0	0
	1	1
Substracción	0	1
	1	0

Tabla 1. *Determinación del signo A_0 después de cada desplazamiento.*

Ejemplo 4. *Multiplicar el multiplicando* $X = 10110$ (-10) *por el multiplicador* $Y = 10001$ (-15).
El detalle de la multiplicación se da en la figura 24. El resultado es:

$$XY = 0010010110 \quad (+150).$$

Se observará en este ejemplo que cuando $C = 1$ la operación anterior es una adición y el último bit $Q_5 = 1$. La operación siguiente se convierte en una substracción seguida de un desplazamiento.

$Q_4\ Q_5$	Operación anterior	Nueva operación	Acumulador	Registro Q	C
			0 | 0000	1 | 0001	5
01	Adición	Sum. X	1 | 0110		
			1 | 0110	1 | 0001	5
		Desp. dcha.	1 | 1011	0 | 1000	4
0	Adición	Desp. dcha.	1 | 1101	1 | 0100	3
0	Adición	Desp. dcha.	1 | 1110	1 | 1010	2
0	Adición	Desp. dcha.	1 | 1111	0 | 1101	1
1	Adición	Substr. X	1 | 0110		
			0 | 1001	0 | 1101	1
		Desp. dcha. (Resul.)	0 | 0100	1 | 0110	0

Fig. 24. *Detalle de la multiplicación del ejemplo* 4.

Observación. Se puede producir un desbordamiento de capacidad del acumulador durante la multiplicación. En este caso los desplazamientos a la derecha se efectúan invirtiendo el signo del acumulador. Un test sobre la naturaleza de la operación efectuada y el signo del multiplicando permite prever todos los casos que se presentan en la práctica. Si no hay desbordamiento de capacidad los desplazamientos se efectúan conservando el signo del acumulador. Este caso se ha tratado en el ejemplo 3.

15.3 Multiplicación multidigital

Por multiplicación multidigital se entiende la multiplicación simultánea de varios dígitos del multiplicador por el multiplicando. Hemos visto anteriormente que la multiplicación binaria era únicamente una sucesión de sumas con desplazamiento pero de uno de los productos parciales. Es así como hemos visto, que si el multiplicador es 0 ó 1 hay o no hay suma de multiplicando. Se puede generalizar esta idea considerando no sólo un dígito sino dos dígitos del multiplicador. Dos dígitos del multiplicador dan cuatro posibilidades 00, 01, 10, 11. Examinemos sucesivamente cada una de estas posibilidades suponiendo un multiplicando $|X| = 1010$. En el caso en que los dígitos del multiplicador son 00, no tendremos

adición del multiplicando o, más exactamente, la primera suma parcial tendrá por valor:

0 0 0 0	Primer producto parcial
0 0 0 0	Segundo producto parcial
0 0 0 0 0 0	Suma parcial = 0

En el caso en que los dos dígitos del multiplicador sean 01 tenemos:

1 0 1 0	Primer producto parcial
0 0 0 0	Segundo producto parcial
0 1 0 1 0	Suma parcial = $\mid X \mid$

En este caso efectuaremos la adición del multiplicando $\mid X \mid$ a la suma parcial anterior. Si ahora los dígitos del multiplicador son 10, tenemos:

0 0 0 0	Primer producto parcial
1 0 1 0	Segundo producto parcial
1 0 1 0 0	Suma parcial $\mid 2X \mid$

En este caso se añadirá dos veces el multiplicando $2X$ a la suma anterior. Finalmente si los dígitos del multiplicando son 11 tendremos:

1 0 1 0	Primer producto parcial $\mid X \mid$
1 0 1 0	Segundo producto parcial $\mid 2X \mid$
1 1 1 1 0	Suma parcial : $\mid 3X \mid$

Se añadirá tres veces el multiplicando $\mid 3X \mid$ a la suma anterior. Cada suma parcial se desplazará dos órdenes a la derecha, pues es el resultado de dos productos parciales correspondientes a dos dígitos del multiplicador. Si, previamente, se dispone en dos registros (o en memoria) de los valores $\mid 2X \mid$ $\mid 3X \mid$ se disminuye el número de sumas y se aumenta la velocidad de la multiplicación. La figura 25 da el algoritmo de la multiplicación multidigital.

Dígitos del multiplicador	Operaciones
00	Sin adición del multiplicando
01	Adición del multiplicando, Desp. dcha. 2 órdenes
10	Adición de dos veces el multiplicando, Desp. dcha. de dos niveles
11	Adición de tres veces el multiplicando, Desp. dcha. de dos órdenes

Fig. 25. *Algoritmo de la multiplicación multidígitos.*

El signo se trata separadamente, es decir, los números se representan en valor absoluto y en signo.

Ejemplo 1. *Multiplicar* (+ 13) = 0 | 1 1 0 1 *por* (– 6) = 1 0 1 1 0.

$\lvert X \rvert$	1 1 0 1	
$\lvert Y \rvert$	0 1 1 0	
$\lvert 2X \rvert$	1 1 0 1 0	
Desp. dcha. 2 ord.	1 1 0 1 0	
Adición $\lvert X \rvert$	1 1 0 1	
	1 0 0 1 1 1 0	
Desp. dcha. 2 ord.	1 0 0 1 1 1 0	
Introducción signo XY	| 0 1 0 0 0 1 1 0	Resultado (– 78)

Ejemplo 2. *Multiplicar* (+ 46) = 0 1 0 1 1 1 0 *por* (– 57) = 1 1 1 1 0 0 1

$\lvert X \rvert$	1 0 1 1 1 0	$\lvert X \rvert$ = 1 0 1 1 1 0
$\lvert Y \rvert$	1 1 1 0 0 1	$\lvert 2X \rvert$ = 1 0 1 1 1 0 0
$\lvert X \rvert$	1 0 1 1 1 0	$\lvert 3X \rvert$ = 1 0 0 0 1 0 1 0
Desp. dcha. 2 ord.	1 0 1 1 1 0	
Suma $\lvert 2X \rvert$	1 0 1 1 1 0 0	
	1 1 0 0 1 1 1 1 0	
Desp. 2 ord.	1 1 0 0 1 1 1 1 0	
Suma $\lvert 3X \rvert$	1 0 0 0 1 0 1 0	
	1 0 1 0 0 0 1 1 1 1 1 0	
Desp. de 2 ord.	1 0 1 0 0 0 1 1 1 1 1 0	
Introd. signo	1 | 0 0 1 0 1 0 0 0 1 1 1 1 1 0	(– 2622)

16. La división automática

Al ser la división un proceso inverso a la multiplicación, su ejecución en ordenador deberá tener alguna semejanza con ella. En efecto, en la división, el dividendo es la suma del producto del cociente por el divisor más el resto si existe.

Se distinguen dos métodos, según que los números a dividir se representen en valor absoluto y en signo o en complemento a 2 si son negativos: el método llamado de restauración de los restos y el método de comparación de los signos de los restos y del divisor. Vamos a examinar los dos métodos sucesivamente.

16.1 División por el método de restauración de los restos parciales

16.1.1 *Caso de los números superiores a 1*

Se procede como en el caso de la división normal restando sucesivamente el módulo del divisor de los órdenes más elevados del módulo del dividendo y de los restos sucesivos, previamente desplazados un orden a la izquierda. Se restaura el dividendo cada vez que la substracción del divisor dé un resto parcial negativo. Al comienzo de la operación el dividendo está contenido, con su signo (A_1) en el acumulador y registro Q, el divisor con su signo (R_1) en el registro R. El signo del resultado se obtiene antes del comienzo de la división y se guarda en memoria mediante una báscula de signo B. Sgn ($A_1 \oplus R_1 \Rightarrow B$. Sgn).

El acumulador tiene una posición intermedia A_0 destinada a llevar un informe de las operaciones de suma y resta. El cociente se realiza orden por orden en el registro Q. El signo del cociente se introduce al final de la división en Q_1 (B Sgn $\Rightarrow$ Q_1).

El contenido del acumulador y del registro Q puede desplazarse un orden a la izquierda. Después de cada substracción, el acumulador contiene los restos parciales de la división. Un contador de desplazamientos C contiene, desde el comienzo de la operación, el número de desplazamientos necesarios para efectuar la división. El conjunto de los registros se da en la fig. 26 en el caso de un cociente limitado a cuatro órdenes más signo, es decir cuyo contenido no puede ser superior a 15 en valor absoluto. En el caso en que la división dé origen a un cociente superior a 15 en valor absoluto el resultado en Q será inexacto. Este caso de desbordamiento de capacidad se detectará desde la primera operación. En lo que continúa consideraremos A_0 como una posición con signo. La división comienza por la substracción del módulo del divisor del contenido del acumulador. Se obtiene así el primer resto. Si es positivo ($A_0 = 0$) se termina la división. El dividendo es demasiado elevado respecto al divisor lo que supone un desbordamiento de capacidad del registro Q. Para poder efectuar la operación se necesita que el primer resto sea negativo ($A_0 = 1$). Se restaura el dividendo añadiendo el módulo del divisor al primer resto, y se desplaza un orden a la izquierda el contenido del acumulador y del registro Q. Se procede a continuación de la siguiente forma: se resta sistemáticamente el módulo del divisor del contenido del acumulador. Si el resto es negativo ($A_0 = 1$), el primer dígito del cociente es 0 y se introduce en Q_5 ($0 \Rightarrow Q_5$) y la operación siguiente es una suma del divisor para restaurar el contenido del acumulador. Se desplaza a continuación el contenido del acumulador y del registro Q un orden a la izquierda. Si el resto es positivo ($A_0 = 0$), el dígito del cociente es 1 ($1 \Rightarrow Q_5$), se desplaza inmediatamente un orden a la izquierda y la operación siguiente es una resta del divisor. Se continúa así hasta el momento en que el contenido del contador de desplazamiento, que disminuye una unidad a cada desplazamiento, es cero. El acumulador contiene entonces un último resto y el resto Q el cociente. Si el último resto es negativo se restaura el contenido del acumulador

sumando una última vez el divisor. El acumulador contiene entonces el resto exacto de la división. El signo del resto se da por el signo del dividendo (A_1). Se recordará que el signo del resto es siempre igual al del dividendo. Examinemos algunos ejemplos:

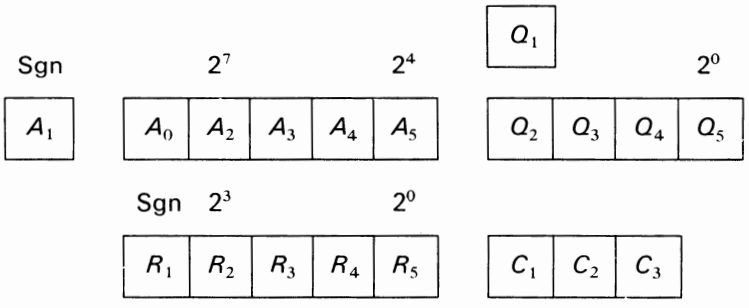


Fig. 26. *Registros utilizados en la división por el método de restauración de restos.*

Ejemplo 1. *Dividir* $+14$ *por* -5.
El cociente es -2 y el resto $+4$. El detalle de la división se da en la figura 27.

C	A_1	$A_0\ A_2\ A_3\ A_4\ A_5$	$Q_2\ Q_3\ Q_4\ Q_5$	$R_1\ R_2\ R_3\ R_4\ R_5$	Observaciones
4	0	0 0 0 0 0	1 1 1 0	1 0 1 0 1	Comienzo de la operación
	0	0 0 0 0 0	1 1 1 0	0 1 0 1	$A_1 \oplus R_1 \Rightarrow$ B Sgn
	0	− 0 1 0 1			Substracción
	0	① 1 0 1 1	1 1 1 0	0 1 0 1	Resto negativo
	0	+ 0 1 0 1			Adición
	0	0 0 0 0 0	1 1 1 0	0 1 0 1	Restauración
3	0	0 0 0 0 1	1 1 0 0	0 1 0 1	Desp. izq. (AQ)(*)
	0	− 0 1 0 1			Substracción
	0	① 1 1 0 0	1 1 0 0	0 1 0 1	Resto $< 0, 0 \Rightarrow Q_5$
	0	+ 0 1 0 1			Adición
	0	0 0 0 0 1	1 1 0 0	0 1 0 1	Restauración
2	0	0 0 0 1 1	1 0 0 0	0 1 0 1	Desp. izq. (AQ)
	0	− 0 1 0 1			Substracción
	0	① 1 1 1 0	1 0 0 0	0 1 0 1	Resto $< 0, 0 \Rightarrow Q_5$
	0	+ 0 1 0 1			Adición
	0	0 0 0 1 1	1 0 0 0	0 1 0 1	Restauración
1	0	0 0 1 1 1	0 0 0 0	0 1 0 1	Desp. izq. (AQ)
	0	− 0 1 0 1			Substracción
	0	⓪ 0 0 1 0	0 0 0 1	0 1 0 1	Resto $> 0, 1 \Rightarrow Q_5$
0	0	0 0 1 0 0	0 0 1 0	0 1 0 1	Desp. izq. (AQ)
	0	− 0 1 0 1			Substracción
	0	① 1 1 1 1	0 0 1 0	0 1 0 1	Resto $< 0, 0 \Rightarrow Q_5$
	0	+ 0 1 0 1			Adición
	0	0 0 1 0 0	0 0 1 0	0 1 0 1	Restauración
	0	0 0 1 0 0	1 | 0 0 1 0	0 1 0 1	$A_1 \oplus R_1 \Rightarrow Q_1$
		$R_n = r_n\, 2^{-n}$	Cociente		(B Sgn $\Rightarrow Q_1$)
		Resto $= +4$	$= -2$		

Fig. 27. *Detalle de la división de* $+14$ *por* -5.

(*) Desp. izq. (AQ) : desplazamiento a la izquierda del acumulador y el registro Q

Ejemplo 2. *Dividir* – 70 *por* + 6.

El cociente es – 11 y el resto – 4 (Fig. 28).

C	A_1	$A_0\ A_2\ A_3\ A_4\ A_5$	$Q_2\ Q_3\ Q_4\ Q_5$	$R_1\ R_2\ R_3\ R_4\ R_5$	Observaciones
4	1	0 0 1 0 0	0 1 1 0	0 0 1 1 0	Comienzo de la operación
		0 0 1 0 0	0 1 1 0	0 1 1 0	$A_1 \oplus R_1 \Rightarrow$ B Sgn
		– 0 1 1 0			Substracción
		① 1 1 1 0	0 1 1 0	0 1 1 0	Resto < 0
		+ 0 1 1 0			Adición
		0 0 1 0 0	0 1 1 0	0 1 1 0	Restauración
3		0 1 0 0 0	1 1 0 0	0 1 1 0	Desp. izq. (AQ)
		– 0 1 1 0			Substracción
		⓪ 0 0 1 0	1 1 0 1	0 1 1 0	Resto > 0, 1 ⇒ Q_5
2		0 0 1 0 1	1 0 1 0	0 1 1 0	Desp. izq. (AQ)
		– 0 1 1 0			Substracción
		① 1 1 1 1	1 0 1 0	0 1 1 0	Resto < 0, 0 ⇒ Q_5
		+ 0 1 1 0			Adición
		0 0 1 0 1	1 0 1 0	0 1 1 0	Restauración
1		0 1 0 1 1	0 1 0 0	0 1 1 0	Desp. izq. (AQ)
		– 0 1 1 0			Substracción
		0 0 1 0 1	0 1 0 1	0 1 1 0	Resto > 0, 1 ⇒ Q_5
0		0 1 0 1 0	1 0 1 0	0 1 1 0	Desp. izq. (AQ)
		– 0 1 1 0			Substracción
		⓪ 0 1 0 0	1 0 1 1	0 1 1 0	Resto > 0, 1 ⇒ Q_5
	1	0 0 1 0 0	1 | 1 0 1 1	0 1 1 0	$A_1 \oplus R_1 \Rightarrow Q_1$
		Resto = – 4	Cociente =		(B Sgn ⇒ Q_1)
		$R_n = r_n\, 2^{-n}$	– 11		

Fig. 28. *Detalle de la división de* – 70 *por* + 6.

16.2 Observación respecto al último resto

La interpretación del último resto exige algunos comentarios. En efecto, éste aparece en el acumulador, es decir, en los órdenes 2^4 a 2^7, debe ser siempre inferior al divisor que está comprendido entre los órdenes 2^0 y 2^3. Para obtener la lectura exacta del último resto es necesario multiplicar éste por 2^{-n}, donde n es el número de dígitos del divisor y del cociente. Así en el ejemplo 1 obtenemos si r_n es el último resto $r_n = 64 = 2^6$ atribuyendo al r_n los órdenes del acumulador. El resto real R_n será $R_n = r_n\, 2^{-n}$, es decir $R_n = r_n\, 2^{-4} = 64.2^{-4} = 4$. De una forma general tendremos:

$$R_n = r_n\, 2^{-n}.$$

16.1.2 *Caso de números inferiores a* 1

El método es idéntico si los números a dividir son inferiores a 1. Sólo los órdenes de los registros no son los mismos; esta vez son potencias negativas de 2 (Fig. 29). Damos un ejemplo:

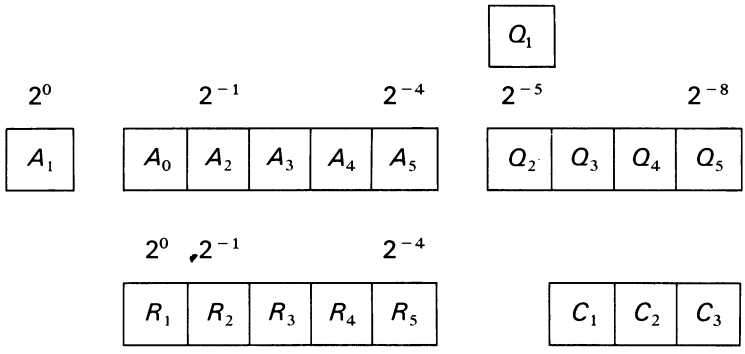


Fig. 29. *Ordenes de los registros en el caso de la división de dos números inferiores a* 1.

Ejemplo. *Dividir* – 107/256 *por* 10/16.
El resultado es – 100/256 y un resto – 7/256 (Fig. 30).

C	A_1	$A_0\ A_2\ A_3\ A_4\ A_5$	$Q_2\ Q_3\ Q_4\ Q_5$	$R_1\ R_2\ R_3\ R_4\ R_5$	Observaciones
4	1	0 0 1 1 0	1 0 1 1	0 1 0 1 0	Comienzo de la operación
		0 0 1 1 0	1 0 1 1	1 0 1 0	$A_1 \oplus R_1 \Rightarrow$ B Sgn
		– 1 0 1 0			Substracción
		① 1 1 0 0	1 0 1 1	1 0 1 0	Resto < 0
		+ 1 0 1 0			Adición
		0 0 1 1 0	1 0 1 1	1 0 1 0	Restauración
3		0 1 1 0 1	0 1 1 0	1 0 1 0	Desp. izq. (AQ)
		– 1 0 1 0	0 1 1 0	1 0 1 0	Substracción
		⓪ 0 0 1 1	0 1 1 1	1 0 1 0	Resto > 0, 1 ⇒ Q_5
		0 0 1 1 0	1 1 1 0	1 0 1 0	Desp. izq. (AQ)
		– 1 0 1 0			Substracción
		① 1 1 0 0	1 1 1 0	1 0 1 0	Resto negativo, 0 ⇒ Q_5
	1	+ 1 0 1 0			Adición
		0 0 1 1 0	1 1 1 0	1 0 1 0	Restauración
1		0 1 1 0 1	1 1 0 0	1 0 1 0	Desp. izq. (AQ)
		– 1 0 1 0			Substracción
		⓪ 0 0 1 1	1 1 0 1	1 0 1 0	Resto > 0, 1 ⇒ Q_5
0		0 0 1 1 1	1 0 1 0	1 0 1 0	Desp. izq. (AQ)
		– 1 0 1 0			Substracción
		① 1 1 0 1	1 0 1 0	1 0 1 0	Resto < 0, 0 ⇒ Q_5
		+ 1 0 1 0			Adición
	1	0 0 1 1 1	1 0 1 0	1 0 1 0	Restauración $A_1 \oplus R_1 \Rightarrow Q_1$
	1	0 0 1 1 1	1 1 0 1 0	1 0 1 0	(B Sgn ⇒ Q_1)
		Resto = 7/256	$Q = -\frac{100}{256}$		
		$R_n = r_n\, 2^{-n}$			

Fig. 30. *Detalle de la división* – 107/256 *por* 10/16

16.3 División por el método de comparación de los signos de los restos parciales y del divisor

Es el método que se utiliza más ampliamente en los ordenadores. Utiliza la representación en completo a 2 de los números negativos. El dígito signo se considera como un dígito ordinario y no se restauran los restos parciales si son negativos. Estos se desplazan sistemáticamente un orden a la izquierda y la operación siguiente así como cada dígito del cociente se impone por comparación del signo de los restos parciales y del divisor. Al comienzo de la operación el dividendo está en el acumulador y el resto Q con su signo en la posición A_1. El divisor está en el registro R con su signo en la posición R_1 y permanece en este registro durante toda la duración de la división. Un contador C, registra al comienzo de la operación el número de desplazamientos necesarios para efectuar la división. La figura 31 da la representación de los diferentes registros utilizados en la división. Se supone que el divisor y el cociente tienen cinco posiciones, una de ellas con signo; el dividendo tiene diez posiciones, una de ellas con signo.

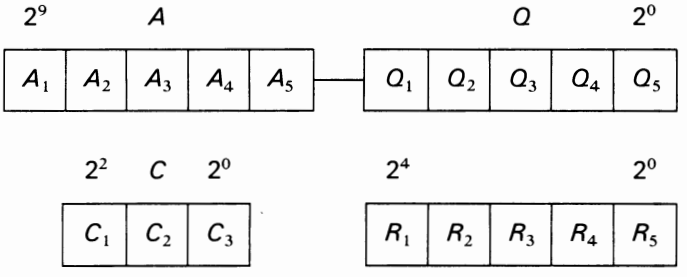


Fig. 31. *Disposición de los registros en la división. El dividendo está dispuesto en el conjunto de los registros A y Q con el signo en la posición A_1. El divisor está en el registro R con el signo en la posición R_1.*

El algoritmo de la división *en el caso de los números superiores a* 1, es el siguiente: Se desplaza el dividendo contenido en el conjunto $A - Q$ *un orden a la izquierda* y se disminuye una unidad al contenido del contador de desplazamientos. Si los signos del dividendo y del divisor antes del desplazamiento son idénticos, la siguiente operación es restar el divisor del contenido del acumulador. Si son diferentes la siguiente operación es la suma del divisor al contenido del acumulador. Se obtiene así el primer resto parcial. La operación continúa comparando el signo del primer resto parcial al del divisor. Si son idénticos, el primer dígito del cociente es 1 ($1 \rightarrow Q_5$) y la siguiente operación después del desplazamiento de un orden a la izquierda del primer resto parcial y del cociente es una resta del divisor del contenido del acumulador. Si son opuestos, el primer dígito del cociente es 0 ($0 \rightarrow Q_5$) y la operación siguiente después del desplazamiento del primer resto

parcial y del cociente es una suma del divisor al contenido del acumulador. Se continúa así por aproximaciones hasta el momento en que se obtiene el penúltimo resto parcial. Un último desplazamiento seguido de una última operación (suma y resta del divisor según los signos del penúltimo resto y del divisor) dando entonces el último resto. El contenido del contador de desplazamientos es en este caso nulo. Se observará que el cociente se elabora orden a orden en el registro Q. Al final de la división el registro Q contiene el cociente con su signo en la posición Q_1 y el acumulador el resto de la división. El resultado que se obtiene así será exacto si el resto es nulo y si el dividendo y el divisor son positivos o si el dividendo es negativo y el divisor es positivo. En los otros casos será necesario corregir el último resto o el cociente o los dos a la vez y efectuar un test final para examinar la validez de la operación.

16.3.1 *Corrección del resto*

Cada vez que el signo del último resto de la división es opuesto al del dividendo se corregirá su valor añadiéndole o quitándole *sin desplazamiento previo* el valor del divisor según que los signos del último resto y del divisor sean diferentes o idénticos. En ciertos casos el último resto no es nulo aunque la división da un resto nulo. Este caso se obtiene detectando en el curso de la división un resto idénticamente nulo (ver más adelante).

16.3.2 *Corrección del cociente*

Se necesita una corrección del cociente en todos los otros casos que se han señalado anteriormente, es decir cuando:

1° El dividendo es positivo y el divisor negativo.
2° El dividendo es negativo y el divisor positivo salvo si el resto es nulo.
3° Dividendo negativo, divisor negativo y resto nulo.

En todos estos casos se añade 1 al cociente. Esto equivale a corregir el cociente cuando el signo del resto es distinto al signo del divisor (*).

16.3.3 *Caso de invalidez de la división*

Son evidentes un cierto número de casos de invalidez de la división:

– *divisor* nulo: eliminado antes del comienzo de la operación,

(*) En efecto, en los dos primeros casos, para un mismo signo del resto parcial, los dígitos del cociente son opuestos a los que se obtendrían con operandos positivos. El cociente negativo se elabora bajo la forma de su complemento a 1. Hay que efectuar una corrección por + 1 para obtener el cociente bajo la forma de su complemento a 2. En el tercer caso el cociente positivo es inferior en una unidad al cociente real. Esto se origina en el valor cero, asignado al último orden del cociente cuando el divisor está contenido en el penúltimo resto, pues el último resto es nulo. Es necesario añadir igualmente 1 al resultado.

– *dividendos demasiado grandes* negativos, lo que corresponde a las configuraciones comprendidas entre los límites siguientes:

1 0 0 0 0 0 0 0 0 0

1 0 1 1 1 1 1 1 1 1

Estas divisiones se eliminan antes del comienzo de la operación mediante examen de los dos primeros órdenes del acumulador $A_1 A_2 = 10$.

Si el divisor es positivo se eliminarán los dividendos correspondientes a $A_1 A_2 = 01$.

– *desbordamiento de capacidad del registro cociente.*

Según que el divisor sea negativo o positivo la invalidez de la operación se controla por la presencia de un resto más allá del signo del acumulador en la primera operación con la condición de que el primer resto no sea nulo, o por comparación del signo del cociente al final de la división con el obtenido con la regla de los signos.

Observación. No se tienen en cuenta los desbordamientos de capacidad del acumulador en el transcurso de la operación salvo para la primera adición o substracción que designe un desbordamiento del registro cociente cuando el divisor es negativo, así como lo hemos visto anteriormente. Damos ahora algunos ejemplos. Se aplicarán las reglas dadas anteriormente.

Ejemplo 1. *Dividir* + 12 *por* + 5.

Los dígitos rodeados de círculos representan los signos del dividendo de los restos parciales y del divisor, y los dígitos subrayados los sucesivos órdenes del cociente (Fig. 32).

C	Acumulador	Registro Q	Divisor	Observaciones
5	⓪0 0 0 0	0 1 1 0 0	⓪0 1 0 1	Comienzo de la divis.
4	0 0 0 0 0	1 1 0 0 0	0 0 1 0 1	Desp. izq. (AQ)
	— 0 0 1 0 1			Substracción
	①1 0 1 1	1 1 0 0 [0]	0 0 1 0 1	$0 \Rightarrow Q_5$
3	1 0 1 1 1	1 0 0 0 0	0 0 1 0 1	Desp. izq. (AQ)
	+ 0 0 1 0 1			Adición
	①1 1 0 0	1 0 0 [0 0]	0 0 1 0 1	$0 \Rightarrow Q_5$
2	1 1 0 0 1	0 0 0 0 0	0 0 1 0 1	Desp. izq. (AQ)
	+ 0 0 1 0 1			Adición
	①1 1 1 0	0 0 [0 0 0]	0 0 1 0 1	$0 \Rightarrow Q_5$
1	1 1 1 0 0	0 0 0 0 0	0 0 1 0 1	Desp. izq. (AQ)
	+ 0 0 1 0 1			Adición
	⓪0 0 0 1	0 [0 0 0 1]	0 0 1 0 1	$1 \Rightarrow Q_5$
0	0 0 0 1 0	0 0 0 1 0	0 0 1 0 1	Desp. izq. (AQ)
	— 0 0 1 0 1			Substracción
	①1 1 0 1	[0 0 0 1 0]	0 0 1 0 1	$0 \Rightarrow Q_5$
	+ 0 0 1 0 1			El signo del resto es diferente del signo del dividendo. Corrección del resto mediante la suma del divisor sin desplazamiento. Sin corrección del cociente.
	[0 0 0 1 0]	[0 0 0 1 0]	0 0 1 0 1	
	Resto = + 2	Cociente = + 2		

Fig. 32. *Detalles de la división de* + 12 *por* + 5.

Ejemplo 2. *Dividir* $+12$ *por* -5.

Después de lo visto, tendremos en todos los casos una corrección del cociente y eventualmente una corrección del resto si éste tiene un signo opuesto al del dividendo (Fig. 33).

C	Acumulador	Registro Q	Divisor	Observaciones
5	⓪0 0 0 0	0 1 1 0 0	①1 0 1 1	Comienzo de la operación
4	0 0 0 0 0	1 1 0 0 0	1 1 0 1 1	Desp. izq. (AQ)
	+ 1 1 0 1 1			Adición
	①1 0 1 1	1 1 0 0 1	1 1 0 1 1	$1 \Rightarrow Q_5$
3	1 0 1 1 1	1 0 0 1 0	1 1 0 1 1	Desp. izq. (AQ)
	− 1 1 0 1 1			Substracción
	①1 1 0 0	1 0 0 1 1	1 1 0 1 1	$1 \Rightarrow Q_5$
2	1 1 0 0 1	0 0 1 1 0	1 1 0 1 1	Desp. izq. (AQ)
	− 1 1 0 1 1			Substracción
	①1 1 1 0	0 0 1 1 1	1 1 0 1 1	$1 \Rightarrow Q_5$
1	1 1 1 0 0	0 1 1 1 0	1 1 0 1 1	Desp. izq. (AQ)
	− 1 1 0 1 1			Substracción
	⓪0 0 0 1	0 1 1 1 0	1 1 0 1 1	$0 \Rightarrow Q_5$
0	0 0 0 1 0	1 1 1 0 0	1 1 0 1 1	Desp. izq. (AQ)
	+ 1 1 0 1 1			Adición
	①1 1 0 1	1 1 1 0 1	1 1 0 1 1	$1 \Rightarrow Q_5$
	1 1 0 1 1	+ 1		Signo del resto diferente del signo del dividendo. Corrección del resto mediante substracción del dividendo sin desplazamiento. Corrección del cociente mediante suma de + 1.
	0 0 0 1 0	1 1 1 1 0	1 1 0 1 1	
	Resto = + 2	Cociente = − 2		

Fig. 33. *Detalles de la división de* $+12$ *por* -5.

Ejemplo 3. *Dividir* – 12 *por* –5.

El dividendo y el divisor son negativos. El cociente sólo se corregirá cuando el último resto sea nulo. El resto se corregirá si su signo es diferente del del dividendo (Fig. 34).

C	Acumulador	Registro Q	Divisor	Observaciones
5	ⓘ1 1 1 1	1 0 1 0 0	ⓘ1 0 1 1	Comienzo de la operación
4	1 1 1 1 1	0 1 0 0 0	1 1 0 1 1	Desp. izq. (AQ)
	– 1 1 0 1 1			Substracción
	⓪0 1 0 0	0 1 0 1 [0]	1 1 0 1 1	$0 \Rightarrow Q_5$
3	0 1 0 0 0	1 0 0 0 0	1 1 0 1 1	Desp. izq. (AQ)
	+ 1 1 0 1 1	1 0 0 0 0	1 1 0 1 1	Adición
	⓪0 0 1 1	1 0 0 [0 0]	1 1 0 1 1	$0 \Rightarrow Q_5$
2	0 0 1 1 1	0 0 0 0 0	1 1 0 1 1	Desp. izq. (AQ)
	+ 1 1 0 1 1			Adición
	⓪0 0 1 0	0 0 [0 0 0]	1 1 0 1 1	$0 \Rightarrow Q_5$
1	0 0 1 0 0	0 0 0 0 0	1 1 0 1 1	Desp. izq. (AQ)
	+ 1 1 0 1 1			Adición
	ⓘ1 1 1 1	0 [0 0 0 1]	1 1 0 1 1	$1 \Rightarrow Q_5$
0	1 1 1 1 0	0 0 0 1 0	1 1 0 1 1	Desp. izq. (AQ)
	– 1 1 0 1 1			Substracción
	⓪0 0 1 1	[0 0 0 1 0]	1 1 0 1 1	$0 \Rightarrow Q_5$
	+ 1 1 0 1 1			
	1 1 1 1 0	[0 0 0 1 0]	1 1 0 1 1	Signo del resto diferente del del dividendo. Corrección del resto por suma del divisor sin desplazamiento. Resto $\neq 0$ sin corrección del cociente
	Resto = – 2	Cociente = + 2		

Fig. 34. *Detalles de la división de* – 12 *por* – 5.

Ejemplo 4. *Dividir* -12 *por* $+5$.

Tendremos una corrección del cociente en todos los casos salvo si el resto de la división es nulo. El resto se corregirá si su signo es opuesto al del dividendo (Fig. 35).

C	Acumulador	Registro Q	Divisor	Observaciones
5	⓪1 1 1 1	1 0 1 0 0	⓪0 1 0 1	Comienzo de la divis.
4	1 1 1 1 1	0 1 0 0 0	0 0 1 0 1	Desp. izq. (AQ)
	+ 0 0 1 0 1			Adición.
	⓪0 1 0 0	0 1 0 0 1	0 0 1 0 1	$1 \Rightarrow Q_5$
3	0 1 0 0 0	1 0 0 1 0	0 0 1 0 1	Desp. izq. (AQ)
	− 0 0 1 0 1			Substracción
	⓪0 0 1 1	1 0 0 1 1	0 0 1 0 1	$1 \Rightarrow Q_5$
2	0 0 1 1 1	0 0 1 1 0	0 0 1 0 1	Desp. izq. (AQ)
	− 0 0 1 0 1			Substracción
	⓪0 0 1 0	0 0 1 1 1	0 0 1 0 1	$1 \Rightarrow Q_5$
1	0 0 1 0 0	0 1 1 1 0	0 0 1 0 1	Desp. izq. (AQ)
	− 0 0 1 0 1			Substracción
	⓪1 1 1 1	0 1 1 1 0	0 0 1 0 1	$0 \Rightarrow Q_5$
0	1 1 1 1 0	1 1 1 0 0	0 0 1 0 1	Desp. izq. (AQ)
	+ 0 0 1 0 1			Adición
	⓪0 0 1 1	1 1 1 0 1	0 0 1 0 1	$1 \Rightarrow Q_5$
	0 0 1 0 1			Signo del resto diferente
	1 1 1 1 0	1 1 1 0 1	0 0 1 0 1	del del dividendo. Co-
		+ 1		rrección del resto me-
		1 1 1 1 0		diante substracción del
	Resto (− 2)	Cociente =		divisor. Resto $\neq 0$ co-
		− 2		rrección del cociente por
				suma de + 1.

Fig. 35. *Detalles de la división de* -12 *por* $+5$.

16.3.4 *Caso en que el resto de la división es nulo*

Examinemos ahora el caso en que el resto de la división es nulo, para verificar el método. No hay corrección del resto.

Ejemplo 1. *Dividir* $+30$ *por* $+2$.
El dividendo y el divisor son positivos. Sin corrección del cociente (Fig. 36).

C	Acumulador	Registro Q	Divisor	Observaciones
5	⓪0 0 0 0	1 1 1 1 1	⓪0 0 1 0	Comienzo de la operación
4	0 0 0 0 1	1 1 1 0 0	0 0 0 1 0	Desp. izq. (AQ)
	− 0 0 0 1 0			Substracción
	①1 1 1 1	1 1 1 0 0	0 0 0 1 0	$0 \Rightarrow Q_5$
3	1 1 1 1 1	1 1 0 0 0	0 0 0 1 0	Desp. izq. (AQ)
	+ 0 0 0 1 0			Adición
	⓪0 0 0 1	1 1 0 0 1	0 0 0 1 0	$1 \Rightarrow Q_5$
2	0 0 0 1 1	1 0 0 1 0	0 0 0 1 0	Desp. izq. (AQ)
	− 0 0 0 1 0			Substracción
	⓪0 0 0 1	1 0 0 1 1	0 0 0 1 0	$1 \Rightarrow Q_5$
1	0 0 0 1 1	0 0 1 1 0	0 0 0 1 0	Desp. izq. (AQ)
	− 0 0 0 1 0			Substracción
	⓪0 0 0 1	0 0 1 1 1	0 0 0 1 0	$1 \Rightarrow Q_5$
0	0 0 0 1 0	0 1 1 1 0	0 0 0 1 0	Desp. izq. (AQ)
	0 0 0 1 0			Substracción
	⓪0 0 0 0	0 1 1 1 1	0 0 0 1 0	$1 \Rightarrow Q_5$
	Resto = 0	Cociente = + 15		Resto nulo. Sin corrección del resto. Sin corrección del cociente.

Fig. 36. *Detalles de la división de* $+30$ *por* $+2$.

Ejemplo 2. *Dividir* $+30$ *por* -2.

Estamos en el caso en que el dividendo es positivo y el divisor es negativo. Cualquiera que sea la naturaleza del resto se corregirá siempre el cociente, añadiendo $+1$ (Fig. 37).

C	Acumulador	Registro Q	Divisor	Observaciones
5	⓪0 0 0 0	1 1 1 1 0	①1 1 1 0	Comienzo de la operación
4	0 0 0 0 1	1 1 1 0 0	1 1 1 1 0	Desp. izq. (AQ)
	+ 1 1 1 1 0			Adición
	①1 1 1 1	1 1 1 0 1	1 1 1 1 0	$1 \Rightarrow Q_5$
3	1 1 1 1 1	1 1 0 1 0	1 1 1 1 0	Desp. izq. (AQ)
	− 1 1 1 1 0			Substracción
	⓪0 0 0 1	1 1 0 1 0	1 1 1 1 0	$0 \Rightarrow Q_5$
2	0 0 0 1 1	1 0 1 0 0	1 1 1 1 0	Desp. izq. (AQ)
	+ 1 1 1 1 0			Adición
	⓪0 0 0 1	1 0 1 0 0	1 1 1 1 0	$0 \Rightarrow Q_5$
1	0 0 0 1 1	0 1 0 0 0		Desp. izq. (AQ)
	+ 1 1 1 1 0			Adición
	⓪0 0 0 1	0 1 0 0 0		$0 \Rightarrow Q_5$
0	0 0 0 1 0	1 0 0 0 0		Desp. izq.
	+ 1 1 1 1 0			Adición
	⓪0 0 0 0	1 0 0 0 0		$0 \Rightarrow Q_5$
		+ 1		Resto nulo sin corrección. Dividendo + Divisor. − .Corrección del cociente.
	Resto = 0	1 0 0 0 1		
		Cociente = − 15		

Fig. 37. *Detalles de la división de* $+30$ *por* -2.

Ejemplo 3. *Dividir* -30 *por* $+2$.

El dividendo es negativo, el divisor positivo y el resto nulo. No se corregirá ni el cociente ni el resto (Fig. 38).

C	Acumulador	Registro Q	Divisor	Observaciones
5	①1 1 1 1	0 0 0 1 0	⓪0 0 1 0	Comienzo de la operación
4	1 1 1 1 0	0 0 1 0 0	0 0 0 1 0	Desp. izq. (AQ)
	+ 0 0 0 1 0			Adición
	⓪0 0 0 0	0 0 1 0 1	0 0 0 1 0	$1 \Rightarrow Q_5$
3	0 0 0 0 0	0 1 0 1 0	0 1 0 1 0	Desp. izq. (AQ)
	− 0 0 0 1 0			Substracción
	①1 1 1 0	0 1 0 1 0	0 0 0 1 0	$0 \Rightarrow Q_5$
2	1 1 1 0 0	1 0 1 0 0	0 0 0 1 0	Desp. izq. (AQ)
	+ 0 0 0 1 0			Adición
	①1 1 1 0	1 0 1 0 0	0 0 0 1 0	$0 \Rightarrow Q_5$
1	1 1 1 0 1	0 1 0 0 0	0 0 0 1 0	Desp. izq. (AQ)
	+ 0 0 0 1 0			Adición
	①1 1 1 1	0 1 0 0 0	0 0 0 1 0	$0 \Rightarrow Q_5$
0	1 1 1 1 0	1 0 0 0 0	0 0 0 1 0	Desp. izq. (AQ)
	+ 0 0 0 1 0			Adición
	⓪0 0 0 0	1 0 0 0 1	0 0 0 1 0	$1 \Rightarrow Q_5$
	Resto = 0	Cociente = − 15		Resto nulo: sin corrección. Dividendo negativo, divisor positivo y resto nulo; no corrección del cociente.

Fig. 38. *Detalles de la división de* -30 *por* $+2$.

Ejemplo 4. *Dividir* -30 *por* -2.
El dividendo y el divisor son negativos y el resto es nulo. Se corregirá el cociente (Fig. 39).

C	Acumulador	Registro Q	Divisor	Observaciones
5	①1 1 1 1	0 0 0 1 0	①1 1 1 0	Comienzo de la operación
4	1 1 1 1 0	0 0 1 0 0	1 1 1 1 0	Desp. izq. (AQ)
	− 1 1 1 1 0			Substracción
	⓪0 0 0 0	0 0 1 0 0	1 1 1 1 0	$0 \Rightarrow Q_5$
3	0 0 0 0 0	0 1 0 0 0	1 1 1 1 0	Desp. izq. (AQ)
	+ 1 1 1 1 0			Adición
	①1 1 1 0	0 1 0 0 1	1 1 1 1 0	$1 \Rightarrow Q_5$
2	1 1 1 0 0	1 0 0 1 0	1 1 1 1 0	Desp. izq. (AQ)
	− 1 1 1 1 0			Substracción
	①1 1 1 0	1 0 0 1 1	1 1 1 1 0	$1 \Rightarrow Q_5$
1	1 1 1 0 1	0 0 1 1 0	1 1 1 1 0	Desp. izq. (AQ)
	− 1 1 1 1 0			Substracción
	①1 1 1 1	0 0 1 1 1	1 1 1 1 0	$1 \Rightarrow Q_5$
0	1 1 1 1 0	0 1 1 1 0	1 1 1 1 0	Desp. izq. (AQ)
	− 1 1 1 1 0			Substracción
	⓪0 0 0 0	0 1 1 1 0	1 1 1 1 0	$0 \Rightarrow Q_5$
		+ 1		Resto nulo sin corrección. Dividendo y divisor negativos y resto nulo; corrección del cociente.
	0 0 0 0 0	0 1 1 1 1		
	Resto = 0	Cociente = + 15		

Fig. 39. *Detalles de la división de* -30 *por* -2.

16.3.5 *Examen de un caso particular*

Divisor negativo y cociente correspondiente a la capacidad máxima del registro cociente. Este será, por ejemplo, el caso de la división de + 256 por – 16, y el resto nulo. El dividendo es positivo: habrá corrección de cociente (Fig. 40).

C	Acumulador	Registro *Q*	Divisor	Observaciones
5	⓪1 0 0 0	0 0 0 0 0	①0 0 0 0	Comienzo de la operación
4	1 0 0 0 0	0 0 0 0 0	1 0 0 0 0	Desp. izq. (*AQ*)
	+ 1 0 0 0 0			Adición
	⓪0 0 0 0	0 0 0 0 0	1 0 0 0 0	$0 \Rightarrow Q_5$
3	0 0 0 0 0	0 0 0 0 0	1 0 0 0 0	Desp. izq. (*AQ*)
	+ 1 0 0 0 0			Adición
	①0 0 0 0	0 0 0 0 1	1 0 0 0 0	$1 \Rightarrow Q_5$
2	0 0 0 0 0	0 0 0 1 0	1 0 0 0 0	Desp. izq. (*AQ*)
	– 1 0 0 0 0			Substracción
	①0 0 0 0	0 0 0 1 1	1 0 0 0 0	$1 \Rightarrow Q_5$
1	0 0 0 0 0	0 0 1 1 0	1 0 0 0 0	Desp. izq. (*AQ*)
	– 1 0 0 0 0			Substracción
	①0 0 0 0	0 0 1 1 1	1 0 0 0 0	$1 \Rightarrow Q_5$
0	0 0 0 0 0	0 1 1 1 0	1 0 0 0 0	Desp. izq. (*AQ*)
	– 1 0 0 0 0			Substracción
	①0 0 0 0	0 1 1 1 1	1 0 0 0 0	$1 \Rightarrow Q_5$
	– 1 0 0 0 0			Signo del resto opuesto al del dividendo. Corrección del resto por substracción del divisor: *nuevo resto nulo.* Dividendo positivo, divisor negativo y resto nulo. Corrección del cociente.
	0 0 0 0 0	0 1 1 1 1		
		+ 1		
	Resto = 0	1 0 0 0 0		
		Cociente = – 16		

Fig. 40. *Detalles de la división de* + 256 *por* – 16.

Este caso particular corresponde a un cociente límite que puede representarse en el registro cociente. Se notará la existencia de un resto más allá del signo del acumulador desde la primera operación así como en un cierto número de las siguientes operaciones. *El primer resto parcial es nulo.* Permanecerá nulo para todos los cocientes iguales a – 16. Para los dividendos superiores para los cuales el

cociente será superior a − 16, el primer resto no será nulo. El resultado será inexacto y la división inválida. Este caso se detectará mediante la presencia de un resto más allá del signo del acumulador en la primera operación, coincidiendo con un resto no nulo. En el ejemplo que acabamos de tratar al ser el primer resto parcial nulo no se tendrá en cuenta el resto más allá del signo. Los otros restos no tienen ningún significado particular.

16.3.6 *Divisiones no válidas: desbordamiento de capacidad del registro cociente*

Habrá desbordamiento de capacidad del registro cociente cada vez que el cociente sea superior a + 15 o a − 16 y el resultado de máquina será inexacto. Se presentarán dos casos según que el divisor sea positivo o negativo.

16.3.6.1 *Caso de un divisor negativo*

El desbordamiento del registro cociente se señalará mediante un resto más allá del signo y un resto no nulo desde la primera operación.

Ejemplo 1. *Dividir* +56 *por* −2.

El resultado − 28 no puede contenerse en el registro *Q*. La división no es válida.

C	Acumulador	Registro *Q*	Divisor	Observaciones
5	⓪0 0 0 1	1 1 0 0 0	①1 1 1 0	Comienzo de la operación
4	0 0 0 1 1	1 0 0 0 0	1 1 1 1 0	Desp. izq. (*AQ*)
	+ 1 1 1 1 0			Suma
	↖0 0 0 0 1	1 0 0 0 0		$0 \Rightarrow Q_5$
				Resto más allá del signo del acumulador primer resto parcial no nulo: división no válida.

Fig. 41. *División de* +56 *por* − 2.

Ejemplo 2. *Dividir* – 56 *por* – 2.

El resultado + 28 no puede contenerse en el registro *Q* (Fig. 42).

C	Acumulador	Registro *Q*	Divisor	Observaciones
5	①1 1 1 0	0 1 0 0 0	①1 1 1 0	Comienzo de la operación
4	1 1 1 0 0	1 0 0 0 0	1 1 1 1 0	Desp. izq. (*AQ*)
	– 1 1 1 1 0			Substracción
	①1 1 1 0	1 0 0 0 1	1 1 1 1 0	$1 \Rightarrow Q_5$
				Resto más allá del signo del acumulador. Primer resto parcial no nulo división no válida.

Fig. 42. *División de* – 56 *por* – 2.

16.3.6.2 *Caso de un divisor positivo*

El desbordamiento de capacidad del registro Q se señalará al final de la división mediante un signo del cociente (Q_1) INEXACTO. Se prevé en la máquina un test final que compara el signo del resultado máquina con el que debe esperarse aplicando la regla de los signos al comienzo de la operación. Si los resultados son opuestos, la división es no válida.

Observación importante

En ciertos casos el resto real de la división es nulo, aunque el resto entregado por la máquina no sea nulo. Este caso se detectará mediante la presencia de un resto parcial nulo en el acumulador durante la división. Hay que asegurarse que los restantes dígitos del dividendo son nulos. Estos están en el registro Q. Si es éste el caso se anulará el último resto mediante una puesta a cero del contenido del acumulador. La detección de un resto nulo durante la división se obtendrá poniendo un 1 sobre una báscula que recibirá a continuación los restantes órdenes del dividendo contenidos en el registro Q a cada desplazamiento. Si éstos son 0 el último resto es obligatoriamente nulo. Si no lo es, se pondrá a cero. Si los restantes pesos del dividendo contienen un 1 se pondrá a cero la báscula y el resto final será nulo a menos que se produzca una segunda detección de un *contenido nulo* del acumulador, en cuyo caso el proceso vuelve a comenzar.

Ejemplo. *Dividir* -60 *por* $+2$.
El resultado -30 no puede contenerse en el registro cociente (Fig. 43).

C	Acumulador	Registro *Q*	Divisor	Observaciones
5	①1 1 1 0	0 0 1 0 0	⓪0 0 1 0	Comienzo de la operación
4	1 1 1 0 0	0 1 0 0 0	0 0 0 1 0	Desp. izq. (*AQ*)
	+ 0 0 0 1 0			Adición
	①1 1 1 0	0 1 0 0 0	0 0 0 1 0	$0 \Rightarrow Q_5$
3	1 1 1 0 0	1 0 0 0 0	0 0 0 1 0	Desp. izq. (*AQ*)
	+ 0 0 0 1 0			Adición
	①1 1 1 0	1 0 0 0 0	0 0 0 1 0	$0 \Rightarrow Q_5$
2	1 1 1 0 1	0 0 0 0 0	0 0 0 1 0	Desp. izq. (*AQ*)
	+ 0 0 0 1 0			Adición
	①1 1 1 1	0 0 0 0 0	0 0 0 1 0	$0 \Rightarrow Q_5$
1	1 1 1 1 0	0 0 0 0 0	0 0 0 1 0	Desp. izq. (*AQ*)
	+ 0 0 0 1 0			Adición
	⓪0 0 0 0	0 0 0 0 1	0 0 0 1 0	$1 \Rightarrow Q_5$
0	0 0 0 0 0	0 0 0 1 0	0 0 0 1 0	Desp. izq. (*AQ*)
	− 0 0 0 1 0			Substracción
	①1 1 1 0	0 0 0 1 0	0 0 0 1 0	$0 \Rightarrow Q_5$
	0 0 0 0 0			Puesta a cero del resto
			0 0 0 1 0	que debe ser nulo sin corrección del cociente.
				Signo del cociente opuesto al de la regla de los signos. División no válida.

Fig. 43. *División de* -60 *por* $+2$. *Detalle de la división.*

Ejemplo. *Dividir* + 144 *por* – 12.

El resultado – 12 puede contenerlo el registro cociente. El resto es nulo. Este ejemplo es muy importante pues hace aparecer el caso señalado anteriormente de un resto no nulo elaborado por la máquina mientras que el resto real de la división es nulo. En la columna de la izquierda se ha indicado el estado de la báscula B. Resto que pasa a 1 cuando se detecta un resto parcial nulo durante la división. Si los restantes dígitos del dividendo, son nulos la báscula B. Resto permanecerá en 1, lo que indica que el resto de la división debe ser nulo al final de la división. Si el cociente es positivo, los restantes dígitos del cociente son ceros, si es negativo, los restantes dígitos son 1. En este último caso el cociente se presenta bajo la forma de su complemento a 1. Se efectuará entonces una corrección por + 1 para obtener su valor bajo la forma de su complemento a 2. Esta corrección obedece a las condiciones de comparación entre el signo, previamente puesto a cero, enunciadas anteriormente o a la comparación directa entre el signo del último resto y del divisor.

C	B. Resto	Acumulador	Registro *Q*	Divisor	Observaciones
5	0	⓪0 1 0 0	1 0 0 0 0	①0 1 0 0	Comienzo de la operación
4		0 1 0 0 1	0 0 0 0 0	1 0 1 0 0	Desp. izq. (*AQ*)
		+ 1 0 1 0 0			Adición
	0	①1 1 0 1	0 0 0 0 1	1 0 1 0 0	$1 \Rightarrow Q_5$
3		1 1 0 1 0	0 0 0 1 $\bar{0}$	1 0 1 0 0	Desp. izq. (*AQ*)
		– 1 0 1 0 0			Substracción
	0	⓪0 1 1 0	0 0 0 1 0	1 0 1 0 0	$0 \Rightarrow Q_5$
2		0 1 1 0 0	0 0 1 $\bar{0}$ $\bar{0}$	1 0 1 0 0	Desp. izq. (*AQ*)
		+ 1 0 1 0 0			Adición
	1	⓪0 0 0 0	0 0 1 0 0	1 0 1 0 0	$0 \Rightarrow Q_5$ Resto parcial nulo
1		0 0 0 0 0	0 1 $\bar{0}$ $\bar{0}$ $\bar{0}$	1 0 1 0 0	Desp. izq. (*AQ*) $1 \Rightarrow B$. Resto.
		+ 1 0 1 0 0			Adición
	1	①0 1 0 0	0 1 0 0 1	1 0 1 0 0	$1 \Rightarrow Q_5$
0		0 1 0 0 0	1 $\bar{0}$ $\bar{0}$ 1 $\bar{0}$	1 0 1 0 0	Desp. izq. (*AQ*)
		– 1 0 1 0 0			Substracción
	1	①0 1 0 0	1 $\bar{0}$ $\bar{0}$ 1 1	1 0 1 0 0	$1 \Rightarrow Q_5$
		⓪0 0 0 0	1 $\bar{0}$ $\bar{0}$ 1 1		B. Resto = 1. Puesta a cero
			+ 1		del último resto.
		⓪0 0 0 0	1 0 1 0 0		Dividendo positivo, divisor negativo: corrección
		Resto = 0	Cocie. = – 12		del cociente.

Fig. 43bis. *División de* + 144 *por* – 12. *Detalle de la división.*

16.4 División en el caso en que los números son inferiores a la unidad

En este caso todos los números son inferiores a 1, lo que impone obligatoriamente un divisor mayor que el dividendo. Un test de validación, efectuado al comienzo de la división, comprobará que es correcto. Recordemos que un número binario x inferior a 1 representado en un registro de n órdenes incluido el signo está comprendido entre los siguientes valores:

$$1 - 2^{1-n} \geqslant x \geqslant -(1 - 2^{1-n}),$$

si es positivo y entre

$$1 - 2^{1-n} \geqslant x \geqslant -1$$

si es negativo. Se observará que el desbordamiento de capacidad se efectúa hacia los órdenes inferiores aunque este efecto puede despreciarse generalmente. Se remediará este inconveniente introduciendo un 1 en los órdenes 2^{-n} del registro cociente. Tendremos siempre un resultado aproximado pero el error cometido será tanto más pequeño cuanto mayor número de órdenes comprende el registro cociente.

El algoritmo es el siguiente. Al principio de la división se dispone el dividendo con su signo (2^0) en el acumulador y el registro Q, el divisor en el registro R. Se compara como anteriormente el signo del dividendo y de los restos parciales al signo del divisor. Si los signos son idénticos, el dígito del cociente es 1 y la operación siguiente después del desplazamiento del contenido del acumulador y del registro Q un orden a la izquierda, es la substracción del divisor del contenido del acumulador. Si los signos son diferentes, el dígito del cociente es cero y la siguiente operación, después del desplazamiento, es la adición del divisor al contenido del acumulador. El primer dígito del cociente se introduce inmediatamente en Q_5 y antes del desplazamiento. Se pierde así el último orden del dividendo. El número de desplazamientos necesarios para efectuar la división es $n - 1$ si n es el número de dígitos del cociente, incluido el signo. Se termina la división cuando es nulo el contenido del contador de desplazamiento. El cociente que se obtiene así es un seudocociente. Deberá corregirse de la siguiente forma: invertir el dígito de la posición de signo e introducir un 1 en el último orden del cociente.

Ejemplo. *Dividir* 128/256 *entre* 160/256.
El resultado exacto es 0,8 con un resto nulo (Fig. 44).

C	Acumulador	Registro Q	Divisor	Observaciones
4	⓪1 0 0 0	0 0 0 0 0	⓪1 0 1 0	Comienzo de la operación
	0 1 0 0 0	0 0 0 0 1	0 1 0 1 0	$1 \Rightarrow Q_5$
3	1 0 0 0 0	0 0 0 1 0	0 1 0 1 0	Desp. izq. (AQ)
	− 0 1 0 1 0			Substracción
	⓪0 1 1 0	0 0 0 1 1	0 1 0 1 0	Resto $> 0{,}1 \Rightarrow Q_5$
2	0 1 1 0 0	0 0 1 1 0	0 1 0 1 0	Desp. izq. (AQ)
	− 0 1 0 1 0			Substracción
	⓪0 0 1 0	0 0 1 1 1	0 1 0 1 0	Resto $> 0{,}1 \Rightarrow Q_5$
1	0 0 1 0 0	0 1 1 1 0	0 1 0 1 0	Desp. izq. (AQ)
	− 0 1 0 1 0			Substracción
	①1 0 1 0	0 1 1 1 0	0 1 0 1 0	Resto $< 0, 0 \Rightarrow Q_5$
0	1 0 1 0 0	1 1 1 0 0	0 1 0 1 0	Desp. izq. $(A\bar{Q})$
	+ 0 1 0 1 0			Adición
	①1 1 1 0	1 1 1 0 0	0 1 0 1 0	Resto $< 0, 0 \Rightarrow Q_5$
	+ 0 1 0 1 0			Signo del resto diferente
	0 0 0 0 0	1 1 1 0 0	0 1 0 1 0	del signo del dividendo.
	0 0 0 0 0	0 1 1 0 1		Corrección del resto. Corrección del cociente.
	$R_n = 0$	$Q = 0.825$		$(\bar{A}_1) \Rightarrow A_1$
				$1 \Rightarrow Q_5$

Fig. 44. *División de* 128/256 *entre* 160/256.

16.5 División mediante la elaboración simultánea de varios dígitos del cociente

En los métodos anteriores se elabora cada dígito del cociente después de cada operación: adición o substracción del divisor; mediante examen del *resto o por* comparación de este signo con el del divisor. En el caso del método de restauración de los restos es posible prever cuál será el dígito siguiente del cociente sin efectuar la adición o la substracción. Su efecto, cuando un resto parcial contiene dos ceros sucesivos después del signo se puede decir sin más que los dos dígitos consecutivos del cociente, serán 1 y 0 si el signo es positivo o 0 y 1 si el signo es negativo. Se desplaza entonces el conjunto acumulador-registro Q un orden a la

izquierda sin efectuar la operación siguiente. Se economiza así una adición o substracción cada vez que se detecten dos ceros consecutivos en el resto después del signo.

17. Las calculadoras decimales

Se llaman máquinas decimales, las máquinas que realizan los cálculos en base 10. No obstante, como la máquina sólo distingue dos entradas, cada dígito decimal del 0 al 9 se representará mediante cuatro dígitos binarios. Así, de este modo, el número 1967 se escribirá:

0001	1001	0110	0111
1	9	6	7

Esta forma de representar los números corresponde al código BCD (decimal codificado en binario), todavía llamado código 1.2.4.8. Así para realizar la suma de dos números cuyo resultado sea inferior a 999 habrá que disponer de dos registros de $3 \times 4 = 12$ posiciones cada uno.

Cada grupo de cuatro dígitos se llama *tétrada.* Aunque con 4 dígitos se pueden representar $2^4 = 16$ combinaciones, de las cuales sólo se utilizan 10 (0 a 9). Hay seis combinaciones no válidas (10 a 15) llamadas *seudotétradas;* estas combinaciones pueden aparecer al final de la operación. Se necesita entonces realizar una corrección para obtener el resultado exacto. Este será, por ejemplo, el caso de la adición o de la substracción en que la configuración de ciertos operandos da lugar a seudotétradas.

17.1 Adición y substracción en las máquinas decimales

Las máquinas decimales utilizan, como acabamos de indicar, el código BCD o código 1.2.4.8 para expresar cada uno de los dígitos decimales de los dos operandos (Fig. 45).

Examinemos ahora cómo se representan las seudotétradas en el código BCD *(Fig. 46).*

Se observará que es suficiente añadir a los valores binarios el equivalente $6_{10} = 0110$ para obtener los valores correctos en el código BCD. Igualmente observaremos que la última columna binaria es idéntica a la última columna BCD. *La corrección sólo será necesaria si la posición* 2^3 *del código binario está en* 1 *y una o la otra de las dos posiciones* 2^2 *y* 2^1 *es* 1, *así como si existe un resto más allá del orden más elevado.* La detección de una seudotétrada (suma superior a 10) debe originar naturalmente un resto hacia el orden más débil de la tétrada siguiente. El circuito de corrección en este caso deberá ser:

Decimal	*BCD* 1.2.4.8	
0	0 0 0 0	Tétradas
1	0 0 0 1	
2	0 0 1 0	
3	0 0 1 1	
4	0 1 0 0	
5	0 1 0 1	
6	0 1 1 0	
7	0 1 1 1	
8	1 0 0 0	
9	1 0 0 1	
10	1 0 1 0	Seudotétradas
11	1 0 1 1	
12	1 1 0 0	
13	1 1 0 1	
14	1 1 1 0	
15	1 1 1 1	

Fig. 45. *Código BCD,*

Decimal	Binario	*BCD*	
10	1 0 1 0	0 0 0 1	0 0 0 0
11	1 0 1 1	0 0 0 1	0 0 0 1
12	1 1 0 0	0 0 0 1	0 0 1 0
13	1 1 0 1	0 0 0 1	0 0 1 1
14	1 1 1 0	0 0 0 1	0 1 0 0
15	1 1 1 1	0 0 0 1	0 1 0 1
16	①0 0 0 0	0 0 0 1	0 1 1 0

Fig. 46. *Seudotétradas del código BCD.*

- examinar los tres órdenes superiores del resultado (detección de una seudotétrada),
- verificar la existencia de un resto más allá del orden más elevado del resultado (suma superior a 15),
- corregir eventualmente el resultado.

El algoritmo de la adición de dos dígitos decimales expresados en el código BCD es el siguiente

$$\begin{array}{ll} \text{Resultado} \leqslant 1001 & \text{sin corrección} \\ \text{Resultado} > 1001 & \text{añadir } 0110 \end{array}$$

Examinemos ahora la substracción de dos dígitos decimales. Cuando la diferencia es positiva, no se necesita ninguna corrección.

Si el resultado es negativo, existe un resto más allá del orden más elevado del módulo y éste se expresa como complemento a 2 en el código binario puro.

En el código BCD la diferencia de dos dígitos decimales que da un resultado negativo se expresará mediante el complemento a 10 del módulo del resultado con un resto hacia el orden inmediatamente superior.

Ejemplo. *Sea restar* 8 *de* 7.

El resultado es − 1. Si se efectúa la resta, tenemos:

$$\begin{array}{r} 7 \\ -\ 8 \\ \hline 1 \leftarrow 9 \end{array}$$

y 9 es el complemento a 10 del módulo del resultado de $|1|$. − 1 se expresará en este caso en el código BCD mediante

$$0\,0\,0\,1 \quad 1\,0\,0\,1 \quad (-1)\,.$$

La figura 47 da los equivalentes binarios y BCD de los diez primeros números negativos (− 1 a − 10). Comparando línea a línea se observa que es suficiente restar 0110 del equivalente binario para obtener el resultado exacto en el código BCD. El algoritmo de la substracción de dos dígitos decimales expresados en el código BCD es el siguiente:

$$\begin{array}{ll} \text{Resultado} \geqslant 0 & \text{sin corrección} \\ \text{Resultado} < 0 & \text{suprimir } 0110. \end{array}$$

17.2 Diferentes formas de adición y de substracción en función de la representación de los números negativos

Si los números se representan en valor absoluto y en signo las dos formas de operación son:

- adición directa,
- substracción directa.

La máquina debe efectuar tanto las substracciones como las adiciones. El algoritmo es el mismo que el desarrollado en la figura 8 para los números binarios puros, con la condición de sustituir el complemento a 2 por el complemento a 10.

Si los números negativos se representan por sus complemento a 10 podemos realizar la adición y la substracción bien directamente, bien por adición o substracción del complemento a 10. En estos dos últimos casos la máquina sólo efectúa las adiciones o las substracciones.

Decimal	Binario	*BCD*
– 1	1 | 1 1 1 1	1 | 1 0 0 1
– 2	1 | 1 1 1 0	1 | 1 0 0 0
– 3	1 | 1 1 0 1	1 | 0 1 1 1
– 4	1 | 1 1 0 0	1 | 0 1 1 0
– 5	1 | 1 0 1 1	1 | 0 1 0 1
– 6	1 | 1 0 1 0	1 | 0 1 0 0
– 7	1 | 1 0 0 1	1 | 0 0 1 1
– 8	1 | 1 0 0 0	1 | 0 0 1 0
– 9	1 | 0 1 1 1	1 | 0 0 0 1
– 10	1 | 0 1 1 0	1 | 0 0 0 0

Fig. 47. *Equivalentes binarios y BCD de los* 10 *primeros números negativos.*

Tenemos las cuatro posibilidades siguientes:

- adición directa,
- substracción directa,
- adición por substracción del complemento a 10,
- substracción por adición del complemento a 10.

Se ignora un resto más allá del signo. Si el resultado es positivo aparece directamente, si es negativo, en complemento a 10. El algoritmo es el de la fig. 11, donde se ha sustituido el complemento a 2 por el complemento a 10. Finalmente si los números negativos se representan por sus complementos a 9, tendremos las cuatro posibilidades siguientes:

- adición directa,
- substracción directa,
- adición por substracción del complemento a 9,
- substracción por adición del complemento a 9.

Un resto más allá del signo se añade al orden más inferior de la primera tétrada del resultado. Si el resultado final es positivo aparece directamente, si es negativo mediante el complemento a 9. El algoritmo es el de la figura 9, donde se sustituye el complemento a 1 por el complemento a 9. Realicemos algunos ejemplos.

Adición por complemento a 9 de los números negativos

Ejemplo 1. *Suma de dos números positivos. Se desea sumar los dos números decimales siguientes:* (+289) *y* (+195).

Tenemos una tétrada para representar cada dígito decimal. Cada número decimal se representará por tres tétradas más una posición de signo antes de la tétrada correspondiente al dígito decimal de valor superior. Tenemos entonces:

$$\begin{array}{r} 0.289 \\ +\ 0.195 \\ \hline 0.484 \end{array}$$

El cálculo en la máquina se efectúa según la tabla de la figura 48. Se observará que cuando se está en presencia de una seudotétrada (resultado comprendido entre 10 y 15) se añade $6_{10} = 0110$, lo que se traduce por un valor decimal igual o superior a $16_{10} = 10000$, y por un resto más allá del orden más elevado de cada tétrada. Este mismo resto se obtiene igualmente cuando se añaden dos tétradas iguales o superiores a 8, aunque el resultado no sea una seudotétrada. En todos los casos este resto se elabora antes de la corrección en el momento de la suma de cada tétrada y se dirige hacia la tétrada de orden más elevado. Lo que explica por qué no existe salida resto en el orden más elevado del adicionador del circuito de corrección.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	0	0 0 1 0	1 0 0 0	1 0 0 1
Segundo número	0	0 0 0 1	1 0 0 1	0 1 0 1
Restos		1 ←	1 ←	
Suma	0	0 1 0 0	0 0 1 0	1 1 1 0
Detección de los restos		Resultado sin resto	Resto (suma superior a 16_{10})	Resto (seudo-tétrada)
Corrección adición de 0 ó 0110		0 0 0 0	0 1 1 0	0 1 1 0
Resultado	0	0 1 0 0	1 0 0 0	0 1 0 0
	+	4	8	4

Fig. 48. *Diferentes fases de la adición decimal de* (+289) + (195) = 484.

Ejemplo 2. *Suma de dos números de signo opuesto. Suma positiva. Sumar* + 538 *y* – 143:

$$\begin{array}{rcr} +\ 538 & & 0.538 \\ -\ 143 & \Longrightarrow & 1.856 \\ \hline +\ 395 & & 10.394 \\ & & \llcorner 1 \\ \hline & & 0.395 \end{array}$$

Se añade el complemento a 9 del número negativo al cual se le atribuye un 1 para el signo. Se observa que hay desbordamiento más allá del signo. Este resto más allá del signo se añade al orden inferior. Se obtiene así el resultado 0.395. Al ser positivo el signo del resultado, éste aparece directamente. La tabla de la fig. 49 resume las diferentes fases del cálculo.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	0	0 1 0 1	0 0 1 1	1 0 0 0
Segundo número	1	1 0 0 0	0 1 0 1	0 1 1 0
Restos	1		1	
Suma ①	0	1 1 0 1	1 0 0 1	1 1 1 0
Detección de restos		Resto (seudo-tétrada)	Sin resto	Resto (seudo-tétrada)
Corrección suma de 0 ó 0110		0 1 1 0	0 0 0 0	0 1 1 0 + 1
Resultado	0	0 0 1 1	1 0 0 1	0 1 0 1
	+	3	9	5

Fig. 49. *Diferentes fases de la suma decimal de* (+ 538) + (– 143) = + 395 *complemento a* 9 *de los números negativos.*

Ejemplo 3. *Suma de dos números de signo opuesto. Suma negativa. Sumar* (− 452) + (+ 136) = −316.

Obtenemos, procediendo como anteriormente:

$$\begin{array}{rr} -\ 452 & 1.547 \\ +\ 136 & \Longrightarrow\ 0.136 \\ -\ \overline{316} & \overline{1.683} \Rightarrow -\ 316\,. \end{array}$$

No hay resto más allá del signo. El resultado es negativo y aparece bajo la forma de complemento a 9. Para obtener el resultado real se necesita tomar el complemento a 9 del resultado, lo que da en nuestro caso − 316. Recordemos que sólo se complementa el módulo del número. La tabla de la figura 50 resume las diferentes etapas de la suma.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	1	0 1 0 1	0 1 0 0	0 1 1 1
Segundo número	0	0 0 0 1	0 0 1 1	0 1 1 0
Restos			1 ←	
Suma	1	0 1 1 0	1 0 0 0	1 1 0 1
Detección de restos		Sin resto	Sin resto	Resto (seudo-tétrada)
Corrección suma de 0 ó 0110		0 0 0 0	0 0 0 0	0 1 1 0
Resultado antes de complemento a 9	1	0 1 1 0	1 0 0 0	0 0 1 1
	—	6	8	3
Resultado después del complemento a 9	—	3	1	6

Fig. 50. *Diferentes fases de la suma decimal de* (− 452) + (+ 136) = − 316 *complemento a* 9 *de los números negativos.*

Ejemplo 4. *Suma de dos números negativos. Sea sumar* (– 389) + (– 513) = – 902.
Obtenemos, complementando a 9 los dos números y dando al signo el valor 1

$$\begin{array}{rcr} -\ 389 & & 1.610 \\ -\ \underline{513} & \Longrightarrow & \underline{1.486} \\ -\ 902 & & 11.096 \\ & & \underline{\qquad\quad 1} \\ & & 1.097 \Rightarrow -\ 902\,. \end{array}$$

Se observará que hay un resto más allá del signo. Este resto se suma *antes de la detección de los restos,* en el orden más inferior del resultado. La tabla de la figura 51 da las siguientes etapas del cálculo.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	1	0 1 1 0	0 0 0 1	0 0 0 0
Segundo número	1	0 1 0 0	1 0 0 0	0 1 1 0
Resto	1			
Suma ①	1	1 0 1 0	1 0 0 1	0 1 1 0
Detección de los restos		Resto (seudo-tétrada)	Sin resto	Sin resto
Corrección suma de 0 ó 0110		0 1 1 0	0 0 0 0	0 0 0 0 + 1
Resultado antes de complemento a 9	1	0 0 0 0	1 0 0 1	0 1 1 1
	—	0	9	7
Resultado después de complemento a 9	—	9	0	2

Fig. 51. *Diferentes fases de la suma decimal de* (– 389) + (– 513) = – 902. *Complemento a 9 de los números negativos.*

Hemos examinado todos los casos de suma que pueden aparecer.

17.3 Representación del signo en el caso de una suma decimal serie-paralela

Los casos examinados suponen la utilización de un sumador paralelo: la adición se efectúa orden a orden en cada tétrada, habrá tantos sumadores decimales como dígitos en los dos números a sumar. Hemos visto que en este caso el dígito puede representarse por un dígito binario 0 ó 1 según que el número decimal sea positivo o negativo. Por el contrario en una suma paralelo-serie sólo se dispone de un sumador de tétradas. Los dos números a sumar se introducen en el sumador tétrada a tétrada. En este caso se necesita representar el signo mediante una tétrada. Si el signo es positivo la tétrada signo será nula 0000, si es negativo la tétrada signo tendrá el valor decimal 9 = 1001.

17.4 Desbordamiento de capacidad en la suma decimal

Un desbordamiento de capacidad en el sumador decimal se detectará de la misma forma que en un sumador binario: habrá desbordamiento de capacidad si hay resto del módulo en el signo o resto más allá del signo, pero no los dos. En el caso de una suma paralelo-serie, un desbordamiento de capacidad se señalará mediante un resto del módulo en la *tétrada signo,* o más allá de la *tétrada signo,* pero no en las dos. El primer caso corresponde a la suma de dos números positivos, el segundo a la suma de dos números negativos. Otra forma de actuar consiste en verificar la identidad de los dos signos y el *valor decimal* de la tétrada signo. Habrá desbordamiento de capacidad si la tétrada signo toma el valor decimal 1 = 0001 ó 8 = 1000. Estos resultados son sólo válidos cuando los números negativos se representan por sus complementos a 9 y a 10.

17.5 Circuito de complemento a 9 de un circuito paralelo

Así hemos visto que un número negativo caracterizado por un 1 en su posición de signo, se introduce en la máquina en forma de complemento a 9. Se necesita disponer de un circuito que entregue el complemento a 9 del número dispuesto a la entrada. Este circuito se ha estudiado ya como aplicación del álgebra de Boole al estudio de los circuitos de conmutación. Damos, de nuevo, su configuración. *Figura 52.*

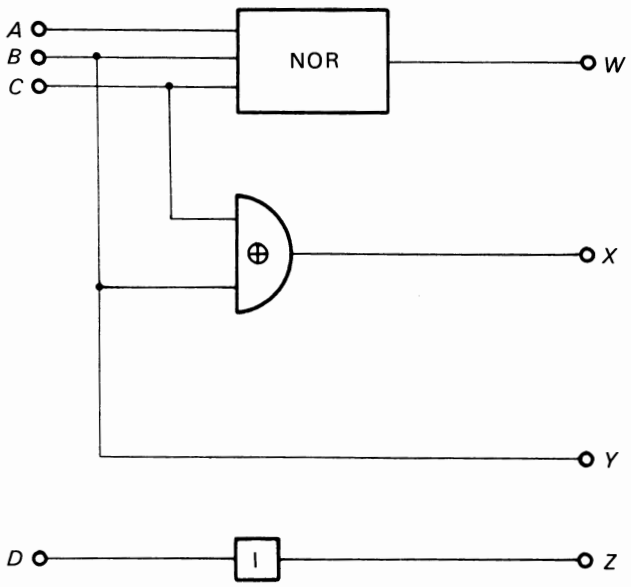


Fig. 52. *Circuito de complemento a* 9.

A la entrada aplicamos una tétrada *ABCD* correspondiente a una cifra decimal comprendida entre 0 y 9. Obtenemos a la salida la tétrada *WXYZ*, que representa el complemento a 9 de la tétrada *ABCD*. Así si la tétrada *ABCD* representa la cifra 3_{10}, es decir 0011, la tétrada *WXYZ*, representa su complemento a 9, es decir, 6_{10} o 0110. Ahora se necesita realizar un circuito que entregue a la salida *ABCD* si el signo es 0 o *WXYZ* si el signo es 1. Este es el circuito de la figura 53.

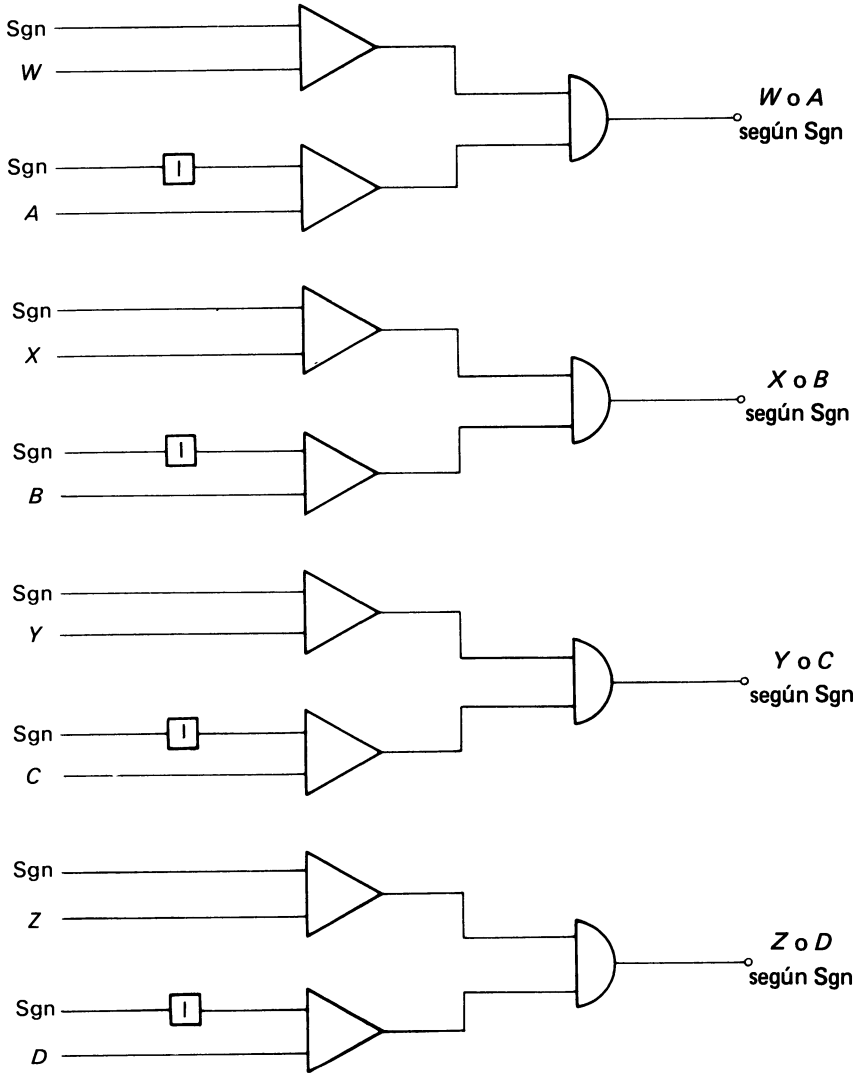


Fig. 53. *Circuito del complemento según el valor del signo.*

17.6 Sumador decimal que utiliza el complemento a 10

Si los números negativos se representan por su complemento a 10 en lugar del complemento a 9, es suficiente suprimir el resto del signo en el orden más inferior del resultado. En efecto, en este caso se ignora un resto más allá del signo. Estamos en el caso del complemento a 9 de una suma binaria. Volvamos al ejemplo 2 anterior.

Suma de dos números de signo opuesto. Suma positiva

$$\begin{array}{rcl} +\ 538 & & 0.538 \\ -\ 143 & \Longrightarrow & 1.857 \\ +\ \overline{395} & & \textcircled{1}\,\overline{0.395} \Longrightarrow +\ 395 \end{array}$$

↓

resto ignorado

El complemento a 10 del módulo de – 143 es el complemento a 9 (856) aumentado en 1 (857). Se ignora el resto más allá del signo. Siendo 0 el signo del resultado, éste es positivo y aparece directamente (+ 395). La tabla de la figura 54 resume las diferentes fases de la suma.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	0	0 1 0 1	0 0 1 1	1 0 0 0
Segundo número	1	1 0 0 0	0 1 0 1	0 1 1 1
Restos	1 ←		1 ←	
Suma 1	0	1 1 0 1	1 0 0 1	1 1 1 1
Detección de los restos		Resto (seudo-tétrada)	Sin resto	Resto (seudo-tétrada)
Corrección adición de 0 ó 0110		0 1 1 0	0 0 0 0	0 1 1 0
Resultado	0	0 0 1 1	1 0 0 1	0 1 0 1
	+	3	9	5

Fig. 54. *Diferentes fases de la suma decimal de* (+ 538) + (– 143) = + 395
Complemento a 10 *de los números negativos. Se ignora el resto más allá del signo.*

Examinemos ahora los otros tres casos.

Suma de dos números de signos opuestos. Suma negativa

Tomemos el ejemplo 3 anterior

$$\begin{array}{rcl} -\ 452 & & 1.548 \\ +\ 136 & \Longrightarrow & \overline{0.136} \\ -\ \overline{316} & & \overline{1.684} \end{array}$$

No hay resto más allá del signo. Este es negativo. Para obtener el resultado exacto hay que tomar el complemento a 10 del módulo de la suma – 316. La tabla (Fig. 55) da las siguientes etapas del cálculo.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	1	0 1 0 1	0 1 0 0	1 0 0 0
Segundo número	0	0 0 0 1	0 0 1 1	0 1 1 0
Restos			1	
Suma	1	0 1 1 0	1 0 0 0	1 1 1 0
Detección de los restos		Sin resto	Sin resto	Resto (seudo-tétrada)
Corrección adición de 0 ó 0110		0 0 0 0	0 0 0 0	0 1 1 0
Resultado antes del complemento a 10	1	0 1 1 0	1 0 0 0	0 1 0 0
	–	6	8	4
Resultado después del complemento a 10	–	3	1	6

Fig. 55. *Diferentes fases de la suma decimal de* (– 452) + (136) = – 316. *Complemento a* 10 *de los números negativos.*

Adición de los números negativos

Volvamos al ejemplo anterior

$$\begin{array}{rcl} -\ 389 & & 1.611 \\ -\ 513 & \Longrightarrow & 1.487 \\ \hline -\ 902 & & \textcircled{1}\ 1.098 \Longrightarrow -\ 902 \end{array}$$

se desprecia
el resto más allá del signo

El complemento a 10 del módulo de – 389 es 611 y del módulo de – 513 es 487. Al ser los dos números negativos su posición de signo es 1. Obtenemos respectivamente 1.611 y 1.487. La suma es decimal para el módulo y binaria para el signo.

Se obtiene 11.098 con un resto más allá del signo. Este se ignora. Al ser el signo negativo, el resultado real es el complemento a 10 de 098, es decir – 902. La figura 56 da las diferentes etapas del cálculo.

Observaciones	Sgn	Tercera tétrada	Segunda tétrada	Primera tétrada
Primer número	1	0 1 1 0	0 0 0 1	0 0 0 1
Segundo número	1	0 1 0 0	1 0 0 0	0 1 1 1
Restos	1			
Suma 1	1	1 0 1 0	1 0 0 1	1 0 0 0
Detección de los restos		Resto (seudo-tétrada)	Sin resto	Sin resto
Corrección adición de 0 ó 0110	1	0 1 1 0	0 0 0 0	0 0 0 0
Resultado antes del complemento a 10	1	0 0 0 0	1 0 0 1	1 0 0 0
	–	0	9	8
Resultado después del complemento a 10	–	9	0	2

Fig. 56. *Diferentes fases de la suma decimal de* (– 389) + (– 513) = – 902. *Complemento a* 10 *de los números negativos.*

Suma de dos números positivos

El resultado es idéntico al del ejemplo del apartado anterior, el lector podrá remitirse a él.

17.7 Circuito para complementar a 10 de un sumando paralelo

El complemento a 10 de un número decimal representado por varias tétradas se obtiene:

1° tomando el complemento a 9 de cada tétrada,
2° añadiendo 1 al orden más inferior de la tétrada que caracteriza las unidades 1 (primera tétrada).

Esta se obtiene mediante el circuito de la figura 52 que proporciona el complemento a 9 de cada tétrada, después, añadiendo 1 al orden más bajo con la ayuda del adicionador paralelo que acabamos de estudiar. El resultado representa el complemento a 10 del número. Aparece en el registro *acumulador* del adicionador. Este número se transferirá a memoria antes de introducirse nuevamente en el adicionador.

18. Multiplicación decimal

Varios métodos permiten la multiplicación de dos números decimales. Citemos entre los más importantes:

- multiplicación por adiciones sucesivas,
- adición del complemento del multiplicando,
- método llamado del "doble del multiplicando y la mitad del multiplicador",
- método de los componentes derecha e izquierda de los productos parciales.

En todos estos métodos los números decimales que se multiplican se representan en valor absoluto y en signo.

18.1 Multiplicación por adiciones sucesivas

En este método cada dígito del multiplicador se toma en memoria en un contador y se le suma tantas veces el multiplicando hasta que quede a cero el contador. A continuación se pasa al segundo dígito decimal, de orden más inferior del multiplicador y se procede como anteriormente, desplazando las sumas parciales un rango decimal.

Ejemplo. *Sea multiplicar* + 531 *por* − 25

|X|	531	
|Y|	25	Contenido del contador : 5
	531	4
	531	3
	531	2
	531	1
	531	0
	2655	
Desp. dcha. 1 rango	2655	Contenido del contador: 2
	531	1
	531	0
	13275	
Introducción del signo	− 13275	(Resultado)

Este método es evidentemente el más simple, pero igualmente es el más largo.

18.2 Variante del método de adiciones sucesivas Reducción del tiempo de la multiplicación

En este caso, cada vez que el dígito decimal del multiplicador es superior a 5, se introduce en un contador que se incrementa en una unidad a cada operación elemental hasta que se pone en 9. Entonces se añade tantas veces el complemento a 10 del multiplicando al contenido del acumulador. Si el dígito precedente del multiplicador fuera inferior a 5, se añade una vez más el complemento a 10 del multiplicando al contenido del acumulador sin cambiar el contenido del contador que continúa con el valor 9. Si el dígito precedente del multiplicador fuese superior a 5 el ciclo suplementario de adición del complemento del multiplicando no se efectúa. El signo se trata por separado. Un resto más allá del orden superior se ignora (complemento a 2). A continuación se desplaza el contenido del acumulador y del registro multiplicador cociente un orden a la derecha. Si el contador marca 9 el contenido del acumulador se representa por su complemento a 10. Hay que introducir un 9 delante del orden más superior del módulo en el momento del desplazamiento de un número negativo representado por su complemento a 2.

Si el dígito del multiplicador es inferior a 5, éste se introduce en el contador que a continuación se disminuye en una unidad a cada operación elemental de la multiplicación. Se añade entonces el multiplicando no complementado al contenido del acumulador hasta el momento en que el contador marque cero. Como anteriormente se realizará una adición suplementaria del multiplicando si el dígito precedente fuese superior a 5. En el caso contrario no hay adición suplementaria. En resumen, la adición suplementaria sólo se realiza cuando se pasa de un dígito del multiplicador inferior a 5 a un dígito siguiente superior a 5 o recíprocamente. Si se pasa de un dígito del multiplicador inferior a 5 a otro dígito inferior a 5 o de un dígito del multiplicador superior a 5 a otro dígito superior a 5 no hay ciclo suplementario de adición del multiplicando. *En este método se dispone el contador delante del módulo del acumulador.* En el momento de cada desplazamiento se añadirá un nueve o un cero delante del módulo del acumulador según su contenido. Finalmente si en el resultado final la posición del contador es 0, el resultado aparece directamente sin corrección. Si el contador marca 9 hay que efectuar un desplazamiento a la derecha suplementario y sumar otra vez el multiplicando para obtener el resultado real.

Ejemplo. *Sea multiplicar* $X = 2684$ *por* $Y = 1862$. *Y es el multiplicador y X el multiplicando.*

Tenemos las siguientes operaciones, observando que cada producto parcial puede contener 5 dígitos. El complemento a 10 de X será en estas condiciones 97316.

	Contador	2684	(X)
		1862	(Y)
$0 \Rightarrow A$, $2 \Rightarrow C$	2	00000	(A)
Suma X, $(C) - 1 \Rightarrow C$		02684	
	1	02684	
Suma X $(C) - 1 \Rightarrow C$		02684	
	0	05368	
Desp. dcha. C y A, $6 \Rightarrow C$	6	005368	
Suma $\overline{X}$, $(C) + 1 \Rightarrow C$		97316	
	7	978528	
Suma $\overline{X}$, $(C) + 1 \Rightarrow C$		97316	
	8	951688	
Suma $\overline{X}$, $(C) + 1 \Rightarrow C$		97316	
	9	924848	
Dígito anterior de $Y < 5$		97316	
Suma $\overline{X}$, $(C) \Rightarrow C$	9	898008	
Desp. dcha. C y A, $8 \Rightarrow C$	8	9898008	
Suma $\overline{X}$, $(C) + 1 \Rightarrow C$		97316	
	9	9629608	
Dígito anterior de $Y > 5$ sin ciclo de adición suplementario			
Desp. dcha. C y A, $1 \Rightarrow C$	1	99629608	
Suma X $(C) - 1 \Rightarrow C$		02684	
	0	02313608	
Dígito anterior de $Y > 5$ ciclo suplementario de adición			
Suma X		02684	
	0	04997608	Resultado

18.3 Método del doble del multiplicando y la mitad del multiplicador

En este método utilizaremos las siguientes fórmulas según la naturaleza del multiplicador

$$XY = 2X\frac{Y}{2} \quad \text{si } Y \text{ es par}$$

$$XY = 2X\left(\frac{Y-1}{2}\right) + X \quad \text{si } Y \text{ es impar.}$$

El resultado de la multiplicación es la suma de los multiplicandos correspondientes a cada multiplicador impar. Se ignorarán los multiplicandos correspondientes a multiplicadores pares.

Ejemplo. *Multiplicar* $X = 52$ *por* $Y = 93$:

$$XY = 52 \times 93 = \underbrace{104 \times 46}_{\Downarrow} + 52$$

$$\underbrace{208 \times 23}_{\Downarrow} + 52$$

$$\underbrace{416 \times 11}_{\Downarrow} + 52 + 208$$

$$\underbrace{832 \times 5}_{\Downarrow} + 52 + 208 + 416$$

$$1664 \times 2 + 52 + 208 + 416 + 832$$

$$3328 \times 1 + 52 + 208 + 416 + 832$$

$$XY = 3320 \times 0 + 52 + 208 + 416 + 832 + 3328$$

$$XY = 4936 .$$

Se observará que los términos de la última suma corresponden a los multiplicandos cuando Y es impar. Es suficiente aplicar las fórmulas anteriores y efectuar la suma de los multiplicandos correspondientes a un multiplicador impar.

18.4 Multiplicación por adición de múltiplos de los dígitos del multiplicando

Llamemos M a cada dígito decimal del multiplicando. Si disponemos de circuitos que realicen el producto de M por 2, 5 y 8 obtenemos para todos los valores posibles de cada dígito decimal del multiplicador P_1 a P_9:

$$P_1 = M , \quad P_2 = 2\,M , \quad P_3 = M + 2\,M ,$$

$$P_4 = 2\,M + 2\,M , \quad P_5 = 5\,M , \quad P_6 = M + 5\,M ,$$

$$P_7 = 2\,M + 5\,M , \quad P_8 = 8\,M , \quad P_9 = M + 8\,M .$$

Esta forma de funcionamiento corresponde a una multiplicación serie que utiliza un sumador decimal de cuatro órdenes. Volveremos en el tomo 2 sobre este tipo de multiplicador.

18.5 Método de los componentes derecha e izquierda de los productos parciales

Cuando se efectúa la multiplicación de dos dígitos decimales el resultado tiene dos dígitos.

Ejemplo.

$$9 \times 8 = 72 , \quad 6 \times 4 = 24 , \quad 1 \times 2 = 02$$

$$4 \times 3 = 12 , \quad 4 \times 2 = 08 .$$

Para efectuar la multiplicación de dos números decimales, que tienen varios dígitos, es suficiente formar los productos parciales reteniendo únicamente los componentes de la derecha y la izquierda sumándoles después del desplazamiento. El signo se trata por separado.

Ejemplo. *Multiplicar* 543 *por* 278

543	*X* Multiplicando
278	*Y* Multiplicador
024	Componente derecha
432	Componente izquierda
4344	Primer producto parcial
581	Componente derecha
322	Componente izquierda
42354	Segundo producto parcial
086	Componente derecha
100	Componente izquierda
150954	Producto *XY*

En este método hay que disponer de una tabla (Fig. 57) que da los componentes derecha e izquierda de todos los productos elementales de todos los números comprendidos entre 0 y 9. Esta tabla se traducirá eléctricamente para obtener los productos parciales de la multiplicación de cada dígito del multiplicando y el multiplicador.

Dígitos del multiplicador	Dígitos del multiplicando	
	Componentes izquierda 0 1 2 3 4 5 6 7 8 9	Componentes derecha 0 1 2 3 4 5 6 7 8 9
0	0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0
1	0 0 0 0 0 0 0 0 0 0	0 1 2 3 4 5 6 7 8 9
2	0 0 0 0 0 1 1 1 1 1	0 2 4 6 8 0 2 4 6 8
3	0 0 0 0 1 1 1 2 2 2	0 3 6 9 2 5 8 1 4 7
4	0 0 0 1 1 2 2 2 3 3	0 4 8 2 6 0 4 8 2 6
5	0 0 1 1 2 2 3 3 4 4	0 5 0 5 0 5 0 5 0 5
6	0 0 1 1 2 3 3 4 4 5	0 6 2 8 4 0 6 2 8 4
7	0 0 1 2 2 3 4 4 5 6	0 7 4 1 8 5 2 9 6 3
8	0 0 1 2 3 4 4 5 6 7	0 8 6 4 2 0 8 6 4 2
9	0 0 1 2 3 4 5 6 7 8	0 9 8 7 6 5 4 3 2 1

Fig. 57. *Componentes derecha e izquierda de un producto de dos dígitos decimales.*

19. La división decimal

La división decimal es similar a la división binaria. Se distinguen dos posibilidades con o sin restauración del dividendo o de los restos parciales. Se procederá de la siguiente forma: restar desde el comienzo el divisor de los dígitos decimales de orden más superior del dividendo, hasta que el resto se hace negativo. El dígito

decimal (cuatro posiciones) de orden superior del cociente es entonces igual al número de restos efectuados.

Esto se obtiene examinando el signo del acumulador. Si éste es positivo después de cada substracción se añadirá una unidad al orden decimal más inferior del registro Q. Cuando el signo del acumulador se hace negativo (1 en la posición del signo) se restaura el resto anterior añadiendo el divisor. Se desplaza un orden decimal (cuatro posiciones a la izquierda) el conjunto acumulador-registro Q. Si existe un resto en el orden más superior se corrige el resultado restando el equivalente de $6_{10} = 0110$.

Se resta a continuación tantas veces Y hasta el momento en que el resto se hace negativo o nulo. En el primer caso se restaura el resto anterior, en el segundo se detiene la división. Si el resto no se hace nulo se seguirá la división hasta la precisión deseada. El método que hemos discutido es el de la restauración del dividendo y de los restos parciales. Realicemos un ejemplo.

Ejemplo. *Dividir* 84_{10} *entre* 7_{10}:

	Sgn	Acumulador		Registro Q			Registro R	
$X \Rightarrow A$	0	1000	0100	0000	0000	0	0000	0111
$Y \Rightarrow R$								
Cuadratura de Y	0	1000	0100	0000	0000	0	0111	0000
Substr. Y	0	0111						
	0	0001	0100					
$A > 0\ (Q) + 1 \Rightarrow Q$	0	0001	0100	0000	0001	0	0111	0000
Substr. Y	0	0111						
	1	1010	0100					
$A < 0\ (Q) + 0 \Rightarrow Q$								
Suma Y	0	0111						
(Rest. + Resta) > 9	0	0001	0100	0000	0001	0	0111	0000
Desp. izq. (AQ)	1	0100	0000	0001	0000	0	0111	0000
Corrección substr. de	0	0110						
$\bar{6}_{10} = 0110$	0	1110	0000					
Substr. Y	0	0111						
$A > 0\ (Q) + 1 \Rightarrow Q$	0	0111	0000	0001	0001	0	0111	0000
Substr. Y	0	0111						
	0	0000	0000					
$A > 0\ (Q) + 1 \Rightarrow Q$	0	0000	0000	0001	0001	0	0111	0000

El resultado aparece en el registro Q 12_{10}. El resto en el acumulador es nulo.

19.1 Utilización de los múltiplos del divisor

En particular $P_2 = 2M$ y $P_5 = 5M$, se procede entonces de la siguiente forma: comenzando por restos 5 M del dividendo, después según que el primer resto sea positivo o negativo se resta o se añade 2 M. La siguiente operación después del desplazamiento de un orden decimal a la izquierda, si el resto anterior era negativo, es una resta de 5 M y se continúa el proceso como se ha indicado. Después de cada desplazamiento a la izquierda se efectúa desde el comienzo una corrección mediante resta de $6_{10} = 0110$ de la segunda tétrada del dividendo como en el ejemplo anterior.

19.2 División por comparación del resto y del divisor

Si se dispone de un dispositivo que compara después de cada substracción, el nuevo resto y el divisor, economizamos una substracción y una suma. En efecto, en cuanto el comparador indica que el resto es menor que el divisor no se substrae éste del último resto. Combinando este método con el anterior se hará una nueva economía de operaciones. Se obtiene una división más rápida, si se dispone en la unidad aritmética simultáneamente de múltiplos del divisor. Disponiendo de nueve circuitos comparados separados es posible determinar cuáles son los múltiplos del divisor que son mayores o más pequeños que el contenido del acumulador, es decir, los órdenes más elevados del dividendo y de los restos parciales. Se obtiene así inmediatamente el cociente de la división y es suficiente la substracción para obtener el resto. Se desplaza entonces el resto un orden decimal a la izquierda y se continúa el proceso hasta la precisión que se desee en el cociente.

20. Cálculos en coma flotante

Los cálculos realizados hasta aquí se realizan sobre dos números en coma fija. Esta se supone a la derecha del número para los enteros superiores a 1 e inmediatamente después del dígito del signo para los números inferiores a 1. Este impone al utilizador el prever a lo largo de los cálculos todos los desplazamientos para poder realizar las operaciones. Al final del cálculo se sitúa el valor real.

Para aligerar el trabajo de programación se realizan máquinas que efectúan los cálculos cualquiera que sea la posición de la coma en un número, dando el resultado con la posición correcta de la coma. Este método lleva el nombre de coma flotante. Todas las calculadoras modernas realizan los cálculos en coma fija o en coma flotante. En este método los números se representan en función de una

potencia de 2, si se trata de números binarios o de una potencia de 10 si son números decimales. Vamos a tratar el método con los números decimales. Es idéntico para los números binarios. Los números se representan bajo la forma

$$X = x\ 10^{\alpha}\ .$$

x es la mantisa del número y α su característica. El número X está *normalizado,* es decir que el primer dígito significativo del módulo se sitúa inmediatamente después de la coma. Esta estará siempre situada a la derecha del signo.

Ejemplo:

$$+\ 3593{,}532 \quad = +\ {,}3593532\ .\ 10^{4}$$
$$-\ {,}002562000 = -\ {,}2562000\ .\ 10^{-2}\ .$$

En general se limita a la expresión de la característica eliminando el factor 10. A continuación se añade una constante al valor del exponente comprendido entre los límites que se le asignan en el registro por un número positivo (por ejemplo 50).

Este nuevo valor se dispone delante del número. Obtenemos entonces:

$$+\ 3593{,}532 \quad = +\ 543593532$$
$$-\ .002562000 \quad = -\ 482562000\ .$$

Detallaremos este punto en el tomo 2.

20.1 Adición y substracción de dos números en coma flotante

Consideremos dos números X e Y tales que

$$X = x\ 10^{\alpha}$$
$$Y = y\ 10^{\beta}\ .$$

Vamos a considerar dos casos según los valores de α y β

1º $\underline{\alpha > \beta}$.

Obtenemos en este caso

$$X \pm Y = (x \pm y\ 10^{-d})\ 10^{\alpha}\ ,$$

con

$$d = \alpha - \beta\ .$$

2º $\underline{\alpha < \beta}$.

Obtenemos:

$$X \pm Y = (x\ 10^{-d} \pm y)\ 10^{\beta}\ .$$

La unidad aritmética efectúa inicialmente la diferencia de las características $d = \alpha - \beta$. Si $d \neq 0$ se desplaza el número que tenga menos características de órdenes a la derecha. La característica superior se hace la característica del resultado. Finalmente éste se normalizará. Hay que prever un posible desbordamiento de las mantisas y las características.

20.2 Multiplicación y división

La multiplicación y división de dos números en coma flotante no necesita su normalización. Tenemos:

$$XY = (xy)\, 10^{\alpha+\beta}\,.$$

Se efectúa el producto de las mantisas y la suma de las características. El producto (xy) será siempre inferior a 1 y eventualmente se necesitará realizar un desplazamiento a la izquierda para normalizar el resultado.

La división se obtendrá de la misma forma:

$$\frac{X}{Y} = \frac{x}{y}\, 10^{\alpha-\beta}\,.$$

Se realiza la división de las mantisas y la diferencia de las características. La división se efectúa por uno de los métodos tratados anteriormente.

21. Diferentes códigos utilizados para representar un número decimal

Además del código 1, 2, 4, 8 ó código B.C.D se pueden utilizar igualmente otros códigos para representar un número decimal. Citemos entre los códigos que se encuentran más frecuentemente los códigos + 3, 2.4.2.1., el código octal y el hexadecimal, el código Gray, el código biquinario.

21.1 Código más 3

Se han buscado códigos que realizan el complemento a 9 por simple permutación de los dígitos, es decir cambiando los "1" por ceros y los "0" por unos. *Este es el caso del código* + 3.

En este código se añade + 3 al número decimal y a continuación se lo codifica en binario. La siguiente tabla da el código + 3 para los nueve primeros dígitos (Fig. 58).

Valor decimal	Valor binario
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

Fig. 58. *Código* + 3.

Complementando cada dígito de cada grupo se obtiene el complemento a 9 del número codificado.

Ejemplo:

3 ⟶ 0 1 1 0

complemento a 1 ⟶ 1 0 0 1 ⟶ 6 .

Para codificar un número decimal mediante el código + 3, se procederá como anteriormente. Se expresará cada dígito decimal mediante su equivalente en el código + 3.

Ejemplo. *Representar el número* 347 *en el código* + 3 *y hallar su complemento.*

Respuesta: 0 1 1 0 | 0 1 1 1 | 1 0 1 0 | .

Si se complementa este número se obtiene

1 0 0 1 | 1 0 0 0 | 0 1 0 1 |

es decir

6 5 2

que es complemento a 9 de 347.

Se observará que el código + 3 no es *ponderado*, pues la suma de los órdenes correspondientes a los dígitos "1" no corresponde al número decimal codificado. Se ha visto anteriormente que el código binario 1.2.4.8. era un código ponderado pero que no permitía obtener el complemento a 9 del número buscado. *Un código que reúne las dos condiciones es el código* 2.4.2.1.

21.2 Código 2.3.2.1

Utiliza siempre cuatro dígitos para representar los números de 0 a 9. Los órdenes afectados a cada dígito según su posición son:

$$2 \quad 4 \quad 2 \quad 1$$

$$a_3 \; a_2 \; a_1 \; a_0 \, .$$

En este caso:

1° La suma de los órdenes de cada dígito unitario da el número buscado.
2° El complemento restringido del número da el complemento a 9 del número decimal.

La siguiente tabla da el código 2.4.2.1 para los nueve primeros dígitos decimales (Fig. 59).

Decimal	2* 4 2 1
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	1 0 1 1
6	1 1 0 0
7	1 1 0 1
8	1 1 1 0
9	1 1 1 1

Fig. 59. *Código* 2* 4 2 1.

21.3 Observaciones respecto al código 2*.4.2.1

Comienza por 0000 y termina en 1111. Todos los códigos ponderados complementarios poseen esta característica. A partir del cinco, y hasta el siete hay dos posibilidades de codificación. Solamente la indicada con 2* permite el complemento de una forma exacta.

Código octal y hexadecimal

El código octal tiene como base el 8, es decir sólo tiene siete símbolos 0, ..., 7. La siguiente tabla da los veinte primeros números decimales con sus equivalentes octales (Fig. 60).

Decimal	Octal	Decimal	Octal
0	0	11	13
1	1	12	14
2	2	13	15
3	3	14	16
4	4	15	17
5	5	16	20
6	6	17	21
7	7	18	22
8	10	19	23
9	11	20	24
10	12		

Fig. 60. *Conversión decimal ⇒ octal.*

21.4 Conversión de un número octal a un número decimal

Ejemplo. *Convertir el número octal* 1213 *en un número decimal:*

$$(1 \times 8^3) + (2 \times 8^2) + (1 \times 8^1) + (3 \times 8^0) = 651_{10}\,.$$

Ejemplo. *Convertir el número octal* 1123 *en número decimal:*

$$(1 \times 8^0) + (1 \times 8^{-1}) + (2 \times 8^{-2}) + (3 \times 8^{-3}) = 1 + 1/8 + 2/64 + 3/512\,,$$

sea:

$$1\,.123_8 = 1\,.83/512 \text{ en base } 10\,.$$

21.5 Conversión de un número octal a binario y recíprocamente

Para convertir un número binario a octal se divide el número binario en grupos de tres dígitos a los que se le atribuye el valor octal correspondiente a la siguiente tabla:

Número octal	Equivalente binario
0	0 0 0
1	0 0 1
2	0 1 9
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Ejemplo. *Convertir* 011101 *en número octal*

Se separa el número binario en grupos de tres dígitos y obtenemos:

$$0\,1\,1 \mid 1\,0\,1 = 35_8 .$$

Otros ejemplos:

$$1\,1\,1 \mid 1\,1\,0 \mid 1\,1\,1_2 = 767_8$$
$$1\,1\,0 \mid 1\,1\,0 \mid 1\,0\,1_2 = 665_8$$

Para convertir un número decimal en octal se le divide sucesivamente por 8 y se agrupan los restos como en el caso de la conversión a binaria:

Ejemplo. *Convertir* 200_{10} *en octal:*

$$\begin{aligned} 200 : 8 &= 25 \rightarrow 0 \uparrow \\ 25 : 8 &= 3 \rightarrow 1 \mid \\ 3 : 8 &= 0 \rightarrow 3 \mid \end{aligned}$$

de donde

$$200_{10} = 310_8 .$$

Ejemplo. *Convertir el número decimal* 3964_{10} *en número octal*:

$$\begin{aligned} 3964 : 8 &= 495 \rightarrow 4 \uparrow \\ 495 : 8 &= 67 \rightarrow 7 \mid \\ 61 : 8 &= 7 \rightarrow 5 \mid \\ 7 : 8 &= 0 \rightarrow 7 \mid \end{aligned}$$

de donde

$$3964_{10} = 7574_8 .$$

Ejemplo. *Convertir el número fraccionario* 0.574_{10} *en su equivalente octal.*

Se procederá aquí como en el caso de un número binario. Se multiplican los números decimales a la derecha de la coma decimal por la base 8 y se mantienen los dígitos a la izquierda de la coma decimal.

$$\begin{aligned} 0.574 \times 8 &= 4.992 \\ 0.992 \times 8 &= 7.936 \\ 0.936 \times 8 &= 7.488 \\ 0.488 \times 8 &= 3.904 \end{aligned}$$

de donde se obtiene:

$$0.574_{10} = 0.4773_8 .$$

Código hexadecimal

El código hexadecimal se da en la figura 61. Se observará que se corresponden las letras de la *A* a la *F* a los valores decimales del 10 al 15.

Binario	Hexadecimal	Decimal
0 0 0 0	0	0
0 0 0 1	1	1
0 0 1 0	2	2
0 0 1 1	3	3
0 1 0 0	4	4
0 1 0 1	5	5
0 1 1 0	6	6
0 1 1 1	7	7
1 0 0 0	8	8
1 0 0 1	9	9
1 0 1 0	*A*	10
1 0 1 1	*B*	11
1 1 0 0	*C*	12
1 1 0 1	*D*	13
1 1 1 0	*E*	14
1 1 1 1	*F*	15

Fig. 61. *Código hexadecimal.*

Para pasar un número binario a hexadecimal se divide el número en grupos de cuatro dígitos, después se aplican los resultados de la tabla anterior.

Ejemplos:

$$1\,0\,1\,1 \mid 1\,0\,1\,1_2 = BB_{16}$$
$$1\,0\,0\,1 \mid 0\,1\,0\,1_2 = 95_{16}$$
$$1\,1\,0\,0 \mid 0\,1\,1\,1_2 = C\,7_{16}$$
$$1\,0\,0\,0 \mid 1\,0\,1\,1_2 = 8\,B_{16}.$$

Inversamente, para pasar un número hexadecimal a decimal se procederá como anteriormente, pero esta vez con base 16.

Ejemplos:

$$AB\ 6_{16} = (10 \times 16^2) + (11 \times 16^2) + (6 \times 16^0) = 2742_{10}.$$

Para los números más grandes se utilizará la siguiente tabla (Fig. 62):

Hexadecimal	Decimal
1 0 0 0	4 0 9 6
2 0 0 0	8 1 9 2
3 0 0 0	1 2 2 8 8
4 0 0 0	1 6 3 8 4
5 0 0 0	2 0 4 8 4
6 0 0 0	2 4 5 7 6
7 0 0 0	2 8 6 7 2
8 0 0 0	3 2 7 6 8
9 0 0 0	3 6 8 6 4
A 0 0 0	4 0 9 6 0
B 0 0 0	4 5 0 5 6
C 0 0 0	4 9 1 5 2
D 0 0 0	5 3 2 4 8
E 0 0 0	5 7 3 4 4
F 0 0 0	6 1 4 4 0

Fig. 62. *Tabla de conversión hexadecimal-decimal para números grandes.*

Ejemplo. *Pasar D 3 A* 6_{16} *a decimal*:

Tendremos:

$$D\ 3\ A\ 6_{16} = D\ 0\ 0\ 0_{16} + 3\ A\ 6_{16}$$

$$D\ 0\ 0\ 0_{16} = 53.248_{10}$$

$$3\ A\ 6 = (3 \times 16^2) + (10 \times 16^1) + (6 \times 16^0) = 934_{10},$$

de donde

$$D\ 3\ A\ 6_{16} = 53.248_{10} + 934_{10} = 54.182_{10}.$$

21.6 Código Gray o código binario reflejado

Hay varios códigos Gray. Todos tienen un punto común. Para cada nuevo valor decimal el equivalente binario no cambia más que un dígito cada vez. El código Gray más conocido es el siguiente (Fig. 63). Se observará que los tres órdenes más débiles son simétricos respecto al aje horizontal que separa el código en dos partes iguales. Por esta razón se llama igualmente código binario simétrico.

Valor decimal	Código Gray
0	0 0 0 0
1	0 0 0 1
2	0 0 1 1
3	0 0 1 0
4	0 1 1 0
5	0 1 1 1
6	0 1 0 1
7	0 1 0 0
8	1 1 0 0
9	1 1 0 1
10	1 1 1 1
11	1 1 1 0
12	1 0 1 0
13	1 0 1 1
14	1 0 0 1
15	1 0 0 0

Fig. 63. *Código Gray.*

21.6.1 *Representación de un número decimal en código Gray*

Para representar un número decimal mediante el código Gray se procede como se indica:

1° Se representa el número decimal con el código binario puro y se pone un 0 a la izquierda del número binario.

2° Después cada dígito del número binario se añade al siguiente mediante suma en módulo dos (o exclusivo).

Ejemplo. *Representar el número* 45 *con el código Gray.*

En binario puro tenemos:

$45_{10} =$ 0 1 0 1 1 0 1

Resultado en código Gray 1 1 1 0 1 1

21.6.2 *Conversión código Gray → decimal codificado en binario*

Consideremos una serie de 0 y de 1 que caracteriza a un número decimal codificado en código Gray. Para obtener un valor decimal, buscaremos primeramente el equivalente del número binario.

Se procederá de la siguiente forma:

- Se parte de los dígitos de órdenes más superiores para ir hacia los dígitos de orden inferior, es decir de izquierda a derecha. Se coloca un 0 a la izquierda del orden más superior.
- Se inscriben los dígitos de orden más superior hasta el momento en que se pasa el primer 1. Se inscribe este primer 1. A partir de este momento sólo se inscribe los 1 hasta el momento en que se encuentra un nuevo 1. Este 1 se sustituye por un cero, así como todos los dígitos siguientes hasta el momento en que se encuentra otro 1, etc. Esto equivale a realizar el o exclusivo entre el 0 y el nivel más superior, después hacer el o exclusivo entre este resultado y el segundo nivel superior, etc.

Ejemplo. *Sea* 001001011 *expresado en código Gray; al pasar a binario queda:*

0 0 1 1 1 0 0 1 0 .

Un código Gray se define mediante *su longitud*, es decir el número de transformaciones que experimenta el número decimal codificado antes de volver a su valor inicial. Se indica entre paréntesis la posición de los órdenes que cambian en cada paso.

Ejemplo. *Establecer el siguiente código Gray:* 9 (1, 2, 1, 3, 1, 2, 1, 3):

0 0 0
0 0 1
0 1 1
0 1 0
1 1 0
1 1 1
1 0 1
1 0 0
0 0 0

La cifra más alta entre paréntesis indica el número más elevado de 1.

Para una longitud determinada pueden existir varios códigos Gray.
Para la longitud 4 sólo existe un código Gray

Longitud	4	(1, 2, 1, 2)
Longitud	6	(1, 2, 1, 3, 2, 3)
		(1, 2, 3, 1, 2, 3)
Longitud	8	(1, 2, 1, 3, 1, 2, 1, 3)
		(1, 2, 3, 4, 1, 4, 3, 2)
Longitud	10	(1, 2, 1, 3, 4, 3, 1, 2, 1, 4)
		(1, 2, 1, 3, 4, 1, 2, 1, 3, 4)

21.7 Código Gray ("rampant")

Es un código Gray de longitud 10 tal que:

Código Gray "rampant": Longitud 10 (1, 2, 3, 4, 5, 1, 2, 3, 4, 5).

La tabla (Fig. 64) da el equivalente binario expresado en código Gray "rampant" de las nueve primeras cifras decimales.

Valor decimal	Código Gray valor decimal
0	0 0 0 0 0
1	0 0 0 0 1
2	0 0 0 1 1
3	0 0 1 1 1
4	0 1 1 1 1
5	1 1 1 1 1
6	1 1 1 1 0
7	1 1 1 0 0
8	1 1 0 0 0
9	1 0 0 0 0

Fig. 64. *Código Gray.*

22. Código biquinario

El código biquinario comprende cinco valores: 0-1-2-3-4-5. Permite contar de 0 a 9. De 0 a 4 el valor 5 vale 0 y de 5 a 9 vale 1. En los cuatro primeros valores un 1 se desplaza a cada valor decimal.

Valor decimal	0 1 2 3 4 5
0	1 0 0 0 0 0
1	0 1 0 0 0 0
2	0 0 1 0 0 0
3	0 0 0 1 0 0
4	0 0 0 0 1 0
5	1 0 0 0 0 1
6	0 1 0 0 0 1
7	0 0 1 0 0 1
8	0 0 0 1 0 1
9	0 0 0 0 1 1

Fig. 65. *Código biquinario.*

El código biquinario es un código ponderado pero no es complementario. La tabla de la figura 65 da el equivalente de las nueve primeras cifras decimales en el código binario. Algunos autores atribuyen el calificativo de biquinario al siguiente código que tiene la propiedad de ser ponderado y sólo utiliza dos 1 para representar los diez símbolos decimales.

Valor decimal	5 0 4 3 2 1 0
0	0 1 0 0 0 0 1
1	0 1 0 0 0 1 0
2	0 1 0 0 1 0 0
3	0 1 0 1 0 0 0
4	0 1 1 0 0 0 0
5	1 0 0 0 0 0 1
6	1 0 0 0 0 1 0
7	1 0 0 0 1 0 0
8	1 0 0 1 0 0 0
9	1 0 1 0 0 0 0

Fig. 66. *Variante del código biquinario.*

Paso del sistema octal al sistema binario

El sistema octal es un sistema de base 8, el sistema binario es de base 2, la relación entre las dos bases es 2^3. La base del sistema octal es la base 2 elevada al cubo. Existe una correspondencia directa entre cada dígito octal y un grupo de los tres dígitos binarios. Un ejemplo hará comprender mejor el mecanismo. Consideremos el siguiente número octal:

$$N_8 = C_2\,8^2 + C_1\,8^1 + C_0\,8^0 .$$

Este mismo número en el sistema binario con nueve dígitos se expresará mediante

$$N_2 = B_8\,2^8 + B_7\,2^7 + B_6\,2^6 + B_5\,2^5 + B_4\,2^4 + B_3\,2^3 + B_2\,2^2 + B_1\,2^1 + B_0\,2^0 .$$

Identifiquemos los dos números poniendo N_2 en forma idéntica a N_8:

$$N_2 = 8^2(B_8\,2^2 + B_7\,2^1 + B_6\,2^0) + 8(B_5\,2^2 + B_4\,2^1 + B_3\,2^0) + 8^0(B_2\,2^2 + B_1\,2^1 + B_0\,2^0)$$

e identificando:

$$C_2 = B_8\,2^2 + B_7\,2^1 + B_6\,2^0\,,$$
$$C_1 = B_5\,2^2 + B_4\,2^1 + B_3\,2^0\,,$$
$$C_0 = B_2\,2^2 + B_1\,2^1 + B_0\,2^0\,.$$

Ejemplo. *Expresar en binario el número octal* 57.

Se expresa directamente cada dígito octal y su equivalente binario mediante tres dígitos:

$$57_8 = 1\ 0\ 1\ 1\ 1\ 1_2\,.$$

El sistema octal puede utilizarse como sistema intermedio en el caso de una conversión de binario a decimal.

Ejemplo. *Consideremos el siguiente número binario:* 111101010. *¿Cuál es su valor decimal?*

Se separa este número en grupos de tres cifras 111 101 010, lo que da su equivalente octal 752 y en valor decimal:

$$7 \times 8^2 + 5 \times 8^1 + 2 \times 8^0 = 490_{10}\,.$$

Bibliografía

Obras generales

AIKEN, *Synthesis of Electronic Computing and Control Circuits*, Harvard Press.

BOOLE, G., *The Mathematical Analysis of Logic.* McMillan, 1848. Nouvelle édition : Blackwell, Oxford, 1948.

BOOLE, G., *An Investigation of the Laws of Thought.* Londres, 1854. Nouvelle édition, Chicago, 1916.

BARON, R., PICCINILLI, A., *Digital Logic and Computer Operations*, 1967.

BOOTH, D. and BOOTH, V., *Automatic Digital Calculators.* Butterworths, 1953.

MCCLUSKEY, *Introduction to the Theory of Switching Circuits.* McGraw Hill, 1965.

CHU, *Digital Computer Design Fundamentals.* McGraw Hill, 1962.

DENIS-PAPIN, M., KAUFMANN, A., FAURE, R., *Cours de calcul booléien appliqué.* Albin Michel, 1963.

HELLERMAN, H., *Digital Computer System Principles*, 1967.

KINTNER, P., *Electronic Digital Techniques.* McGraw Hill, 1968.

KUNTZMANN, J., *Algèbre de Boole.* Dunod, 1969.

LAGASSE, J., ERCEAU, J., *Logique combinatoire et séquentielle.* Dunod, 1969.

LAURET, A., *Principes de programmation des ordinateurs.* Masson, 1967.

PFISTER, *Logical Design of Digital Computers.* Wiley and Sons, 1958.

DU ROSCOAT, J., *Conception de la programmation des ordinateurs.* Masson, 1967.

DEBRAINE, P., *Machines de traitement de l'information.* Tomes 1 et 2, 1967 et 1969.

SOUBIES-CAMY, H., *Les techniques binaires et le traitement de l'information.* Dunod, 1966.

SPEISER, A., *Digitale Rechenanlagen.* Springer Verlag, 1965.

WEYH, U., *Elemente der Schaltungsalgebra.* R. Oldenborng, 1967.

Artículos de interés general

[1] SHANNON, C. E., « A Symbolic Analysis of Relay and Switching Circuits. » *Trans A.I.E.E.*, vol. 57, pp. 713-723, 1938.

[2] SHANNON, C. E., « The Synthesis of Two-terminal Switching Circuits. » *B.S.T.J.*, vol. 28, pp. 59-98. Janvier 1949.

[3] MCCLUSKEY, « Minimization of Boolean Functions. » *B.S.T.J.* Novembre 1956, pp. 1417-1444.

[4] McCLUSKEY, « Logical Design Theory of NOR Gate Networks with no complemented Inputs, Switching Circuit Theory and Logical Design. » *Proc. Fourth Ann. Symposium*, Chicago, 28-30 septembre 1963, pp. 105-116.

[5] KARNAUGH, M., « The Map Method for Synthesis of Combinational Logic Circuits. » *A.I.E.E. Commun. Electronics.* Novembre 1953, pp. 593-599.

[6] VEITCH, E. W., « A Chart Method for Simplifying Truth Functions. » *Proc. Computing Machinery Conf.* 2-3 mai 1952, pp. 593-599.

[7] QUINE, W. V., « A Way to Simplify Truth Functions. » *Am. Math. Monthly*, vol. 62, nº 9, pp. 627-631. Novembre 1955.

[8] BURKS, A. W., GOLDSTINE, H. and VON NEUMANN, J., « Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. » The Institute for Advanced Study, Princeton, N.J., 1947.

[9] LEINER, A. L., SMITH, J. L. and WEINBERGER, A., « System Design of Digital Computer at the National Bureau of Standards. » Natl. Bur. of Standards, Circular 591. Février 1958.

[10] GILCHRIST, B., POMERENE, J. H. and WONG, S. Y., « Fast-carry logic for digital computers. » *IRE Trans. on Electronic Computers*, vol. EC-4, pp. 133-136. Décembre 1955.

[11] LEHMAN, M., « High-speed digital multiplication. » *IRE Trans. on Electronic Computers*, vol. EC-6, pp. 204-205. Septembre 1957.

[12] ROBERTSON, J. E., « A new class of digital division methods. » *IRE Trans. on Electronic Computers*, vol. EC-7, pp. 218-222. Septembre 1958.

[13] BLOCH, E., « The engineering design of the Stretch Computer. » *Proc. EJCC*, Boston, Mass., pp. 48-58. 1-3 décembre 1959.

[14] CAMPBELL, S. J. and ROSSER, G. H. Jr., « An Analysis of Carry Transmission in Computer Addition, » preprints of papers presented at the 13th Natl. Meeting of the ACM, Univ. of Illinois, Urbana. 11-13 juin 1958.

[15] BURTSEV, V. S., « Accelerating Multiplication and Division Operations in High-Speed Digital Computers. » Exact Mechanics and Computing Technique, Acad. Sci. USSR, Moscou, 1958.

[16] ROBERTSON, J. E., « Theory of Computer Arithmetic Employed in the Design of the New Computer at the University of Illinois. » Digital Computer Lab., University of Illinois, Urbana, file nº 319. Juin 1960.

[17] AVIZIENIS, A., « A Study of Redundant Number Representation for Parallel Digital Computers. » Digital Computer Lab., University of Illinois, Urbana, Rept. nº 101. 20 mai 1960.

[18] FRIEMAN, C. V., « A Note on Statistical Analysis of Arithmetic Operations in Digital Computers, » *P.I.R.E.*, janvier 1961, pp. 91-103.

[19] McSORLEY, « High-Speed Arithmetic in Binary Computers. » *P.I.R.E.*, janvier 1961, pp. 67-91.

A. PETITCLERC

TRATADO DE ORDENADORES

TOMO I

ERRATAS

Pág. 27, tabla 1, columna derecha, línea 15; leer: *tiene el valor 0,* en lugar de *tiene el valor 1.*

Pág. 42, fig. 46, columna Símbolo, penúltimo esquema, leer: $A\overline{B}$ en lugar de AB.

Pág. 98, antepenúltima línea, leer: *párrafo 3.7* en lugar de *párrafo 3.6.*

Pág. 104, fig. 83, arriba, a la izquierda, leer: BCD en lugar de ACD.

Pág. 117, fórmula (6), leer: $\overline{S}\,\overline{T}\,\overline{Q}_t$ en lugar de $\overline{S\,T\,Q_t}$.

Pág. 117, fig. 115 (a), leer: Q_{t+1} en lugar de $\overline{Q}_{t+1}$.

Pág. 125, fig. 125: *añadir una cruz en la columna 1 para* $\overline{CD}$.

Pág. 142, fórmula (1): *la cantidad delante de* Σ *debe ser interpretada como un signo.*

Pág. 149, ejemplo 7, el resultado en el acumulador es: $1\,|\,1110$ (-14).

Pág. 150, línea 6, leer: x_0, y_0 en lugar de x_0, y_0, z_0.

Pág. 151, ejemplo 4, leer: $X > 0$, $Y < 0$ en lugar de $X < 0$, $Y > 0$.

Pág. 151, ejemplo 4, línea 5, leer: *resto más allá* en lugar de *sin resto más allá.*

Pág. 152, ejemplo 5, leer: $X < 0$, $Y > 0$ en lugar de $X > 0$, $Y < 0$.

Pág. 154, párrafo 8.2, leer: *resta del orden menos elevado* en lugar de *resta del orden más elevado.*

Pág. 163: *suprimir el último cero a la derecha en la representación (+ 20).*

Pág. 164, párrafo 11.1, Desplazamiento a izquierda: *añadir ceros en las posiciones vacantes.*

Pág. 164, párrafos 11.2 y 11.3, Desplazamiento a izquierda: *añadir 1 en las posiciones vacantes.*

Añadir al final del párrafo 11.3: *La posición signo puede ser incorporada en los desplazamientos de números negativos representados por sus complementos a 2 o a 1 a condición de que se conserve el signo original.*

Pág. 196, fig. 36, Registro Q, línea primera, leer: *11110* en lugar de *11111.*